



**UNIVERSIDAD CARLOS III DE MADRID**

**INGENIERÍA INFORMÁTICA**

# **DISEÑO E IMPLEMENTACIÓN DE UN PORTAL WEB PARA LA ENTREGA DE PRÁCTICAS**

**PROYECTO FIN DE CARRERA**

*Realizado por:*

**Yankoa López Cozas**

*Dirigido por:*

**Javier Fernández Muñoz**

Leganés, Diciembre de 2009



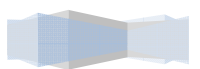
## **AGRADECIMIENTOS**

Dedico este proyecto a mis padres, ya que gracias a ellos he tenido la oportunidad de estudiar una segunda carrera y en todo momento me han mostrado su apoyo y alegría al ver como iba superando todos los retos que me proponía.

También quiero agradecer a mi mejor amigo Josema que siempre haya estado a mi lado soportando mis malos humos y cabreos cuando algo no funcionaba como quería y tenía que quedarme sin salir en casa. Eres un amigo de verdad.

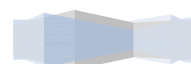
Finalmente agradecer a todos mis profesores durante estos años, todo lo aprendido, a mis nuevos amigos que aunque apenas nos conocemos han estado también apoyándome, y por supuesto a toda mi familia que también siempre ha estado dándome ánimos para continuar.

A todos ellos, gracias.



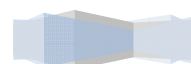
## TABLA DE CONTENIDOS

<b>1.- INTRODUCCIÓN.....</b>	<b>10</b>
1.1.- DESCRIPCIÓN.....	10
1.2.- OBJETIVOS .....	11
1.3.- ACRÓNIMOS Y DEFINICIONES .....	11
1.3.1.- Acrónimos .....	11
1.3.2.- Definiciones.....	13
1.4.- VISIÓN GENERAL DEL DOCUMENTO.....	13
<b>2.- HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.....</b>	<b>15</b>
2.1.- JAVA.....	15
2.1.1.- Introducción.....	15
2.1.2.- Java 2 Platform, Estándar Edition (J2SE).....	16
2.1.3.- Java 2 Platform, Enterprise Edition (J2EE) .....	16
2.1.4.- Java 2 Platform, Micro Edition (J2ME).....	17
2.1.5.- Conclusiones.....	17
2.2.- MYSQL.....	18
2.2.1.- Introducción.....	18
2.2.2.- Utilización de MySQL.....	18
2.2.3.- Tipos de Datos.....	19
2.2.4.- Principales Características .....	20
2.2.5.- Conclusiones.....	21
2.3.- APACHE TOMCAT .....	21
2.3.1.- Introducción.....	21
2.3.2.- Ficheros de Configuración.....	22
2.3.3.- Conclusiones.....	22
2.4.- STRUTS.....	23
2.4.1.- Introducción.....	23
2.4.2.- Conclusiones.....	25
2.5.- HIBERNATE .....	25
2.5.1.- Introducción.....	25
2.5.2.- Conclusiones.....	26
2.6.- JAVASERVER PAGES (JSP) .....	27
2.6.1.- Introducción.....	27
2.6.2.- Sintaxis .....	27
2.6.3.- Librerías de Etiquetas .....	28
2.6.4.- Conclusiones.....	30
2.7.- JAVASCRIPT .....	30
2.7.1.- Introducción.....	30
2.7.2.- Conclusiones.....	31
2.8.- ECLIPSE WEB TOOLS PLATFORM (WTP).....	31
2.8.1.- Introducción.....	31
2.8.2.- Características.....	32
2.8.3.- WTP.....	33
2.8.4.- Conclusiones.....	35
2.9.- OTRAS HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS .....	36
<b>3.- ANÁLISIS DEL SISTEMA.....</b>	<b>37</b>
3.1.- REQUISITOS DE USUARIO .....	37
3.1.1.- Requisitos de Capacidad.....	38
3.1.2.- Requisitos de Restricción .....	41
3.2.- REQUISITOS SOFTWARE.....	44
3.2.1.- Requisitos Software Funcionales.....	45
3.2.2.- Requisitos Software de Interfaz.....	53
3.2.3.- Requisitos Software de Recursos.....	54



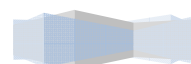


3.2.4.- Requisitos Software de Comprobación.....	55
3.2.5.- Requisitos Software de Documentación .....	57
3.2.6.- Requisitos Software de Seguridad .....	58
<b>4.- DISEÑO DEL SISTEMA .....</b>	<b>59</b>
4.1.- CASOS DE USO .....	59
4.1.1.- Especificación Gráfica de los Casos de Uso.....	59
4.1.2.- Descripción de los Actores .....	61
4.1.3.- Especificación Textual de los Casos de Uso.....	62
4.2.- DIAGRAMA DE CLASES.....	70
4.2.1.- Descripción del Diagrama de Clases .....	72
4.3.- MODELO RELACIONAL .....	76
4.3.1.- Supuestos Semánticos no Recogidos.....	79
4.3.2.- Checks .....	79
<b>5.- IMPLEMENTACIÓN .....</b>	<b>80</b>
5.1.- ARQUITECTURA EN 4 CAPAS .....	80
5.2.- COMUNICACIÓN Y ELEMENTOS DE LAS CAPAS.....	81
5.2.1.- Action (Servlet) .....	83
5.2.2.- La JSP.....	85
5.2.3.- El Manager .....	89
5.2.4.- La Capa de Acceso a Datos (DAO) .....	89
5.2.5.- Hibernate .....	90
5.2.6.- El Paquete de Recursos de Mensajes.....	91
<b>6.- VERIFICACIÓN DEL SISTEMA .....</b>	<b>92</b>
6.1.- PRUEBAS DE CAJA BLANCA.....	92
6.2.- PRUEBAS DE CAJA NEGRA.....	92
6.3.- PRUEBAS DE ACCESIBILIDAD .....	93
6.3.1.- Analizador de Contrastes.....	94
6.3.2.- Validador de Hojas de Estilo .....	95
6.3.3.- Test de Accesibilidad Web .....	96
<b>7.- IMPLANTACIÓN.....</b>	<b>100</b>
7.1.- FORMACIÓN NECESARIA PARA LA IMPLANTACIÓN .....	100
7.2.- INFRAESTRUCTURA .....	101
7.3.- ASPECTOS HARDWARE.....	101
7.4.- ASPECTOS SOFTWARE .....	101
7.4.1.- Java Runtime Enviroment 6 Update 17 (JRE).....	102
7.4.2.- MySQL 5.1 .....	102
7.4.3.- Apache Tomcat 5.5.....	103
7.5.- EJECUCIÓN EN MODO ESTÁNDAR.....	104
7.6.- EJECUCIÓN EN MODO PERSONALIZADO .....	108
7.7.- POSIBLES ERRORES .....	109
<b>8.- ORGANIZACIÓN DEL PROYECTO.....</b>	<b>111</b>
8.1.- ROLES ORGANIZATIVOS Y RESPONSABILIDADES.....	111
8.2.- PLANIFICACIÓN .....	112
8.3.- PRESUPUESTO.....	115
8.3.1.- Cotes de Personal.....	115
8.3.2.- Cotes de Formación.....	115
8.3.3.- Cotes de Material.....	116
8.3.4.- Cote Total .....	117
<b>9.- CONCLUSIONES.....</b>	<b>118</b>
<b>10.- BIBLIOGRAFÍA.....</b>	<b>119</b>
<b>ANEXO I.- MANUALES DE USUARIO .....</b>	<b>121</b>
1.- MANUAL DE USUARIO DEL ADMINISTRADOR.....	121
1.1.- Administrador – Modificar Contraseña .....	122



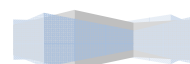


1.2.- Personal Docente - Activar .....	123
1.3.- Personal Docente – Buscar/Modificar .....	125
1.4.- Coordinadores - Activar .....	126
1.5.- Coordinadores – Buscar/Modificar/Borrar .....	127
1.6.- Cerrar Sesión .....	129
2.- MANUAL DE USUARIO DE COORDINADORES Y PROFESORES .....	129
2.1.- Coordinador/Profesor – Modificar Contraseña .....	132
2.2.- Coordinador/Profesor – Cambiar Asignatura .....	133
2.3.- Profesores - Activar .....	133
2.4.- Profesores - Buscar/Modificar/Borrar .....	134
2.5.- Prácticas - Alta .....	136
2.6.- Prácticas – Buscar/Modificar/Borrar .....	139
2.7.- Prácticas - Obtener URL .....	140
2.8.- Prácticas – Recoger Prácticas .....	141
2.9.- Prácticas – Comprobar Firma Digital .....	141
2.10.- Alumnos – Importar/Consultar Listas .....	142
2.11.- Alumnos – Consultar Grupos .....	144
2.12.- Cerrar Sesión .....	145
3.- MANUAL DE USUARIO DE ALUMNOS .....	146
3.1.- Crear Grupo .....	146
3.2.- Modificar Grupo .....	150
3.3.- Entregar Práctica .....	151
3.4.- Cerrar Sesión .....	152
<b>ANEXO II.- IMPLANTACIÓN EN ENTORNOS WINDOWS .....</b>	<b>153</b>
1.- FORMACIÓN NECESARIA PARA LA IMPLANTACIÓN .....	153
2.- INFRAESTRUCTURA .....	153
3.- ASPECTOS HARDWARE .....	153
4.- ASPECTOS SOFTWARE .....	153
4.1.- Java Runtime Enviroment 6 update 17 (JRE) .....	154
4.2.- MySQL 5.1 .....	156
4.3.- Apache Tomcat 5.5 .....	162
5.- EJECUCIÓN EN MODO ESTÁNDAR .....	167
6.- EJECUCIÓN EN MODO PERSONALIZADO .....	168
7.- POSIBLES ERRORES .....	170
<b>ANEXO III.- EL MARCO DE TRABAJO STRUTS .....</b>	<b>171</b>
1.- CONTROLADOR DE STRUTS .....	171
1.1.- La Clase ActionServlet .....	171
1.2.- La Clase Request Processor .....	171
1.3.- La Clase Action .....	172
1.4.- La Clase ActionForward .....	172
2.- MODELO DE STRUTS .....	172
2.1.- Los Objetos de Negocio .....	172
2.2.- La Persistencia .....	173
2.3.- Los Patrones Bussiness Delegate y Data Access Object .....	174
3.- VISTA DE STRUTS .....	174
3.1.- Uso de JavaBeans .....	175
3.2.- La Clase ActionForm .....	175
3.3.- Librerías de Etiquetas .....	175
4.- EL MARCO DE TRABAJO VALIDATOR .....	176
5.- EL MARCO DE TRABAJO TILES .....	177
6.- CONFIGURAR APLICACIONES CON STRUTS .....	178
6.1.- Configurar el Archivo web.xml .....	179
6.2.- Configurar el Archivo struts-config.xml .....	181
<b>ANEXO IV.- EL MARCO DE TRABAJO HIBERNATE .....</b>	<b>185</b>
1.- CONCEPTOS BÁSICOS DE HIBERNATE .....	185
2.- EL FICHERO DE CONFIGURACIÓN DE HIBERNATE .....	185
3.- EL FICHERO DE MAPEO DE HIBERNATE .....	186





3.1.- Mapear Asociaciones.....	187
3.2.- Las Sesiones .....	191
3.3.- Hibernate Query Lenguaje: HQL .....	192



## ÍNDICE DE FIGURAS

FIGURA 1: LOGOTIPO DE JAVA .....	15
FIGURA 2: EDICIONES DE JAVA .....	16
FIGURA 3: LOGOTIPO DE MYSQL .....	18
FIGURA 4: LOGOTIPO DE TOMCAT .....	21
FIGURA 5: MODELO 1 .....	23
FIGURA 6: MODELO 2 .....	24
FIGURA 7: LOGOTIPO DE STRUTS .....	25
FIGURA 8: LOGOTIPO DE HIBERNATE .....	26
FIGURA 9: LOGOTIPO DE ECLIPSE .....	32
FIGURA 10: LOGOTIPO DE ECLIPSE WTP .....	33
FIGURA 11: ALCANCE DE LOS SUBPROYECTOS DE WTP .....	35
FIGURA 12: CASOS DE USO I .....	59
FIGURA 13: CASOS DE USO II .....	60
FIGURA 14: CASOS DE USO III .....	61
FIGURA 15: DIAGRAMA DE CLASES .....	71
FIGURA 16: ARQUITECTURA DEL SISTEMA .....	80
FIGURA 17: MENUADMINISTRADOR.JSP .....	82
FIGURA 18: STRUTS-CONFIG.XML .....	83
FIGURA 19: RECUPERARDOCENTEACTION.JAVA .....	84
FIGURA 20: JSP GUARDAR DOCENTE .....	85
FIGURA 21: ADMINISTRADOR-DEFS.JSP .....	86
FIGURA 22: PLANTILLAADMINISTRADOR.JSP .....	87
FIGURA 23: GUARDARDOCENTE.JSP .....	87
FIGURA 24: GUARDARDOCENTEBODY.JSP .....	88
FIGURA 25: VALIDATION.XML .....	88
FIGURA 26: PERSONALMANAGER.JAVA .....	89
FIGURA 27: DAO.JAVA .....	89
FIGURA 28: PERSONAL.HBM.XML .....	90
FIGURA 29: ANALIZADOR DE CONTRASTES I .....	94
FIGURA 30: ANALIZADOR DE CONTRASTES II .....	95
FIGURA 31: VALIDADOR DE CSS .....	96
FIGURA 32: RESULTADO VALIDADOR DE CSS .....	96
FIGURA 33: TEST DE ACCESIBILIDAD .....	97
FIGURA 34: CONTENIDO DE /ETC/TOMCAT5.5 .....	103
FIGURA 35: CONTENIDO DE /USR/SHARE/TOMCAT5.5 .....	104
FIGURA 36: MYSQL ADMIN I .....	105
FIGURA 37: MYSQL ADMIN II .....	105
FIGURA 38: MYSQL ADMIN III .....	106
FIGURA 39: MYSQL ADMIN VI .....	106
FIGURA 40: DIAGRAMA GANTT I .....	113
FIGURA 41: DIAGRAMA GANTT II .....	114
FIGURA 42: ACCESO AL SISTEMA .....	121
FIGURA 43: INICIO ADMINISTRADOR .....	122
FIGURA 44: MODIFICAR CONTRASEÑA .....	123
FIGURA 45: CONTRASEÑA MODIFICADA .....	123
FIGURA 46: ACTIVAR DOCENTE .....	124
FIGURA 47: DOCENTE ACTIVADO .....	124
FIGURA 48: BUSCAR DOCENTE I .....	125
FIGURA 49: BUSCAR DOCENTE II .....	125
FIGURA 50: BUSCAR DOCENTE III .....	126
FIGURA 51: ACTIVAR COORDINADOR .....	126
FIGURA 52: COORDINADOR ACTIVADO .....	127
FIGURA 53: BUSCAR COORDINADOR I .....	128
FIGURA 54: BUSCAR COORDINADOR II .....	128
FIGURA 55: ACCESO AL SISTEMA .....	129

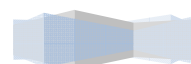






FIGURA 56: SELECCIONAR ASIGNATURA .....	130
FIGURA 57: INICIO COORDINADOR .....	130
FIGURA 58: INICIO PROFESOR CON PERMISOS .....	131
FIGURA 59: INICIO PROFESOR .....	131
FIGURA 60: MODIFICAR CONTRASEÑA .....	132
FIGURA 61: CONTRASEÑA MODIFICADA .....	133
FIGURA 62: ACTIVAR PROFESOR .....	133
FIGURA 63: PROFESOR ACTIVADO .....	134
FIGURA 64: BUSCAR PROFESOR I .....	135
FIGURA 65: R PROFESOR II .....	135
FIGURA 66: R PROFESOR III .....	136
FIGURA 67: ALTA PRÁCTICA .....	136
FIGURA 68: DATOS AUTORES .....	137
FIGURA 69: PRÁCTICA DADA DE ALTA .....	137
FIGURA 70: GUARDAR ARCHIVO .....	138
FIGURA 71: ARCHIVOS DE UNA PRÁCTICA .....	138
FIGURA 72: BUSCAR PRÁCTICA I .....	139
FIGURA 73: BUSCAR PRÁCTICA II .....	139
FIGURA 74: BUSCAR PRÁCTICA III .....	140
FIGURA 75: OBTENER URL .....	140
FIGURA 76: RECOGER PRÁCTICAS ENTREGADAS .....	141
FIGURA 77: COMPROBAR FIRMA DIGITAL .....	142
FIGURA 78: CONSULTAR LISTAS .....	142
FIGURA 79: IMPORTAR LISTA .....	143
FIGURA 80: VISUALIZAR LISTA I .....	144
FIGURA 81: VISUALIZAR LISTA II .....	144
FIGURA 82: CONSULTAR GRUPOS I .....	145
FIGURA 83: CONSULTAR GRUPOS II .....	145
FIGURA 84: INICIO ALUMNOS .....	146
FIGURA 85: CREAR GRUPO .....	147
FIGURA 86: GRUPO CREADO .....	148
FIGURA 87: AÑADIR ALUMNO AL GRUPO .....	148
FIGURA 88: ALUMNO AÑADIDO .....	149
FIGURA 89: MODIFICAR ALUMNO .....	150
FIGURA 90: VALIDACIÓN GRUPOS .....	150
FIGURA 91: ENTREGAR PRÁCTICA I .....	151
FIGURA 92: ENTREGAR PRÁCTICA II .....	152
FIGURA 93: INSTALACIÓN JRE I .....	154
FIGURA 94: INSTALACIÓN JRE II .....	155
FIGURA 95: INSTALACIÓN JRE III .....	155
FIGURA 96: INSTALACIÓN MYSQL I .....	156
FIGURA 97: INSTALACIÓN MYSQL II .....	157
FIGURA 98: INSTALACIÓN MYSQL III .....	157
FIGURA 99: INSTALACIÓN MYSQL IV .....	158
FIGURA 100: INSTALACIÓN MYSQL V .....	158
FIGURA 101: INSTALACIÓN MYSQL VI .....	159
FIGURA 102: INSTALACIÓN MYSQL VII .....	159
FIGURA 103: INSTALACIÓN MYSQL VIII .....	160
FIGURA 104: INSTALACIÓN MYSQL IX .....	160
FIGURA 105: INSTALACIÓN MYSQL X .....	161
FIGURA 106: INSTALACIÓN MYSQL XI .....	162
FIGURA 107: INSTALACIÓN APACHE TOMCAT I .....	163
FIGURA 108: INSTALACIÓN APACHE TOMCAT II .....	163
FIGURA 109: INSTALACIÓN APACHE TOMCAT III .....	164
FIGURA 110: INSTALACIÓN APACHE TOMCAT IV .....	164
FIGURA 111: INSTALACIÓN APACHE TOMCAT V .....	165
FIGURA 112: INSTALACIÓN APACHE TOMCAT VI .....	165
FIGURA 113: INSTALACIÓN APACHE TOMCAT VII .....	166
FIGURA 114: INSTALACIÓN APACHE TOMCAT VIII .....	166
FIGURA 115: CREACIÓN WAR DESDE ECLIPSE .....	170

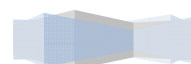
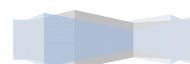


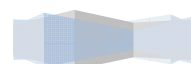


FIGURA 116: COMPONENTES DEL CONTROLADOR DE STRUTS .....	171
FIGURA 117: MOTOR DE PERSISTENCIA .....	173
FIGURA 118: EJEMPLO DE TILES EN UNA APLICACIÓN WEB .....	178
FIGURA 119: RELACIÓN N-M .....	189
FIGURA 120: RELACIÓN 1-N .....	191



## ÍNDICE DE TABLAS

TABLA 1: DATOS NÚMERICOS MYSQL .....	19
TABLA 2: DATOS TIPO FECHA MYSQL.....	20
TABLA 3: DATOS TIPO CADENA MYSQL.....	20
TABLA 4: ELEMENTOS DEL FICHERO SERVER.XML .....	22
TABLA 5: ETIQUETAS JSP .....	28
TABLA 6: ETIQUETAS CORE DE JSTL .....	29
TABLA 7: ETIQUETAS FMT DE JSTL.....	30
TABLA 8: FORMATO DE LOS REQUISITOS DE USUARIO .....	37
TABLA 9: FORMATO DE CASOS DE USO.....	62
TABLA 10: RESULTADOS ACCESIBILIDAD I.....	98
TABLA 11: RESULTADOS ACCESIBILIDAD II .....	99
TABLA 12: COSTES DE PERSONAL I .....	115
TABLA 13: COSTES DE PERSONAL II .....	115
TABLA 14: COSTES DE FORMACIÓN.....	116
TABLA 15: COSTES DE HARDWARE.....	116
TABLA 16: COSTES DE MATERIAL FUNGIBLE .....	116
TABLA 17: OTROS COSTES .....	117
TABLA 18: COSTE TOTAL DEL PROYECTO.....	117
TABLA 19: ELEMENTOS HIJOS DEL ELEMENTO SERVLET.....	179
TABLA 20: ELEMENTOS HIJO DEL ELEMENTO TAGLIB.....	180
TABLA 21: ATRIBUTOS PARA EL ELEMENTO DATA-SOURCE .....	181
TABLA 22: ATRIBUTOS DEL ELEMENTO FORM-BEAN .....	181
TABLA 23: ALGUNOS ATRIBUTOS DEL ELEMENTO ACTION .....	183



## 1.- INTRODUCCIÓN

### 1.1.- DESCRIPCIÓN

Actualmente el sistema de educación universitario español, esta sufriendo diversos cambios, que afectan a la gestión de las titulaciones de las universidades y por su puesto a los contenidos de las asignaturas. Por otro lado, el avance de las nuevas tecnologías y especialmente Internet, hacen que los alumnos interactúan con la universidad por medio de la red facilitando así el acceso a recursos, la realización de matriculas...

Este proyecto está basado en el análisis, diseño y desarrollo, de una aplicación Web destinada a la gestión y entrega de prácticas para las diferentes asignaturas pertenecientes a las titulaciones de una universidad.

Dicha aplicación conseguirá que el personal docente de la universidad pueda gestionar principalmente las asignaturas de una titulación y por lo tanto sus prácticas y alumnos, obteniendo toda la información relevante a ella a través de la red, además de permitir a los alumnos de la universidad realizar la entrega automática de prácticas.

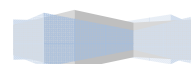
La aplicación esta diseñada para su uso por el administrador de la aplicación y los coordinadores, profesores y alumnos de la universidad. De esta forma diferenciamos cuatro roles:

En primer lugar contaremos con el administrador de la aplicación que tendrá encomendadas diversas tareas mediante las cuales mantendrá la aplicación bajo un correcto funcionamiento y una actualización continua, así, este se encargará entre otras cosas, de instalar la aplicación y configurarla, gestionar a los coordinadores de las asignaturas...

Por un lado, nos encontraremos con el perfil de coordinador de una asignatura. Estos podrán realizar la mayor parte de las acciones de la aplicación, como por ejemplo, gestionar profesores, gestionar prácticas...

Por otro lado, contaremos con los profesores de las asignaturas, dentro de ellos podemos distinguir dos tipos, los profesores con permisos sobre las prácticas, que dada su condición podrán gestionar prácticas, además de realizar las acciones propias de su condición, y los profesores sin permisos que se encargarán de visualizar listados, recoger prácticas...

Finalmente la aplicación podrá ser accedida por los alumnos de la universidad, a través de los enlaces de las diferentes prácticas que son publicados por los profesores. Los alumnos, solamente podrán crear y modificar grupos de prácticas, con la única finalidad de entregar prácticas electrónicamente.



La aplicación se encuentra implementada a través del lenguaje de programación Java en su edición J2EE haciendo uso de los marcos de trabajo Struts e Hibernate. Además utiliza como almacén de datos MySQL, el servidor de Apache Tomcat y diversos patrones de diseño, mediante los cuales se mejora la estructura de la aplicación.

## 1.2.- OBJETIVOS

El principal objetivo de este proyecto, es implementar una aplicación Web que incluya diversas funcionalidades que permitan mejorar la actual aplicación de entrega de prácticas existente en el departamento Arcos de la Universidad Carlos III, con el fin de que pueda ser aplicable a todas las titulaciones de la universidad y su acceso sea el mismo para todos los usuarios, sin distinción de la titulación a la que pertenecen.

Por otro lado, se pretende realizar una aplicación de fácil manejo y que permita modificaciones sencillas de su funcionalidad y arquitectura, debido a los grandes cambios que suceden a menudo en la educación universitaria española.

Finalmente, se desea conseguir que los diferentes usuarios de la aplicación accedan a información dinámica y puedan realizar diferentes tareas utilizando Internet, facilitando así el trabajo de los profesores y coordinadores de las asignaturas, y permitiendo al alumno trabajar desde cualquier lugar de forma segura.

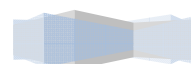
A continuación se detallan los objetivos a cumplir por la aplicación:

- Gestión y administración de coordinadores, profesores, grupos de prácticas y prácticas.
- Importación de listados de alumnos para las diferentes asignaturas.
- Entrega de prácticas y generación de certificados de seguridad.
- Consulta de listados de alumnos.

## 1.3.- ACRÓNIMOS Y DEFINICIONES

### 1.3.1.- ACRÓNIMOS

- **API:** Interfaz de Programación de Aplicaciones (Application Programming Interface).
- **BD:** Base de datos.
- **CU-XX:** Caso de uso, donde XX es un número de dos dígitos.
- **DAO:** Objeto de Acceso a Datos (Data Access Object).





- **HQL:** Hibernate Query Lenguaje.
- **IDE:** Entorno de Desarrollo Integrado.
- **J2EE:** Java 2 Enterprise Edition.
- **J2ME:** Java 2 Mobile Edition.
- **J2SE:** Java 2 Standar Edition.
- **JSTL:** Librería de Etiquetas Estándar de JavaServer Pages (JavaServer Pages Standar Tag Library).
- **JSP:** JavaServer Pages.
- **JVM:** Máquina Virtual de Java.
- **MVC:** Modelo Vista Controlador.
- **ORM:** Mapeo de Objetos Relacionales.
- **POO:** Programación Orientada a Objetos.
- **RSC-XX:** Requisito software de comprobación, donde XX es un número de dos dígitos.
- **RSD-XX:** Requisito software de documentación, donde XX es un número de dos dígitos.
- **RSF-XX:** Requisito software funcional, donde XX es un número de dos dígitos.
- **RSI-XX:** Requisito software de interfaz, donde XX es un número de dos dígitos.
- **RSR-XX:** Requisito software de recursos, donde XX es un número de dos dígitos.
- **RSS-XX:** Requisito software de seguridad, donde XX es un número de dos dígitos.
- **RUC-XX:** Requisito de usuario de capacidad, donde XX es un número de dos dígitos.
- **RUR-XX:** Requisito de usuario de restricción, donde XX es un número de dos dígitos.
- **UML:** Lenguaje Unificado de Modelado (Unified Modeling Language).



- **W3C:** Consorcio para World Wide Web (WWW).
- **WAI:** Iniciativa de Accesibilidad a la Web (Web Accessibility Initiative).
- **WTP:** Web Tool Platform.

### 1.3.2.- DEFINICIONES

---

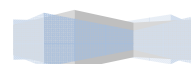
- **JavaDoc:** Utilidad desarrollada por Sun Microsystems que permite comentar clases Java, para la posterior generación de API en formato HTML.
- **MVC:** Patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

## 1.4.- VISIÓN GENERAL DEL DOCUMENTO

---

El actual documento se estructura en los siguientes bloques:

1. **Introducción:** Sección actual en la que se realiza una breve introducción al proyecto, explicando sus principales objetivos.
2. **Herramientas y tecnologías utilizadas:** Describe todas las herramientas y tecnologías empleadas para el correcto desarrollo del proyecto.
3. **Análisis del sistema:** Se describen de forma detallada todos los deseos y necesidades expuestos por el cliente generándose así los requisitos de usuario, para posteriormente refinar dichos requisitos aumentando su nivel de detalle dando lugar a la especificación de requisitos software.
4. **Diseño del sistema:** Muestra el diseño que seguirá el proyecto a través de diferentes diagramas.
5. **Implementación:** Refleja el funcionamiento de la aplicación explicando cada uno de los elementos que la forman y la manera en que estos se comunican para que la información fluya del cliente al servidor.
6. **Verificación del sistema:** Describe brevemente las pruebas llevadas a cabo para verificar el correcto funcionamiento de la aplicación y recoge los pasos seguidos para analizar las páginas y conseguir que fueran accesibles según lo que dicta la W3C.
7. **Implantación:** Plan de implantación seguido para instalar la aplicación en entornos Linux.
8. **Organización del proyecto:** Refleja el tiempo empleado para realizar el proyecto y todos los costes derivados de él.





9. **Conclusiones:** Muestra las conclusiones obtenidas tras la realización del proyecto y las posibles líneas futuras de trabajo.
10. **Bibliografía:** Recoge todos los documentos utilizados para la realización del proyecto.
11. **Anexos:** Información adicional y necesaria para la comprensión del proyecto.





## 2.- HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

Para implementar la aplicación se han seleccionado herramientas y tecnologías que permitiese el funcionamiento multiplataforma, es decir, que la aplicación funcione tanto en Window como en Linux. Por otro lado, también se han elegido herramientas de código abierto con la finalidad de reducir el coste de su proyecto y por supuesto se ha pretendido en todo momento utilizar herramientas que permitan la reutilización y la fácil modificación del código implementado.

A continuación explicaremos brevemente cada una de las herramientas y tecnologías utilizadas.

### 2.1.- JAVA

#### 2.1.1.- INTRODUCCIÓN

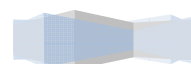
Los lenguajes estructurados se basan en estructuras de control de bloques de código y subrutinas independientes que soportan recursividad y variables locales. La programación orientada a objetos (POO) toma las mejores ideas de la programación estructurada y las combina con nuevos conceptos de organización, permitiendo descomponer un programa en grupos relacionados. Cada subgrupo, pasa a ser un objeto auto contenido con sus propias instrucciones y datos.

Sun describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”.



Figura 1: Logotipo de Java

La existencia de distintos tipos de procesadores y ordenadores, llevó a la conclusión de que era muy importante conseguir un software que no dependiera del tipo de procesador utilizado. Se planteó la necesidad de conseguir código capaz de ejecutarse en cualquier tipo de máquina. Una vez compilado no debería ser necesaria ninguna modificación por el hecho de cambiar de procesador o de ejecutarlo en otra máquina. La clave consistió en desarrollar un código neutro el cual estuviera preparado para ser ejecutado sobre una máquina virtual denominada Java Virtual Machine (JVM).



La JVM es el intérprete de Java, es quien se encarga de ejecutar los ficheros con extensión .class creados por el compilador de Java (javac.exe)

Java ha desarrollado tres ediciones de plataformas diferentes: J2SE, J2EE y J2ME.

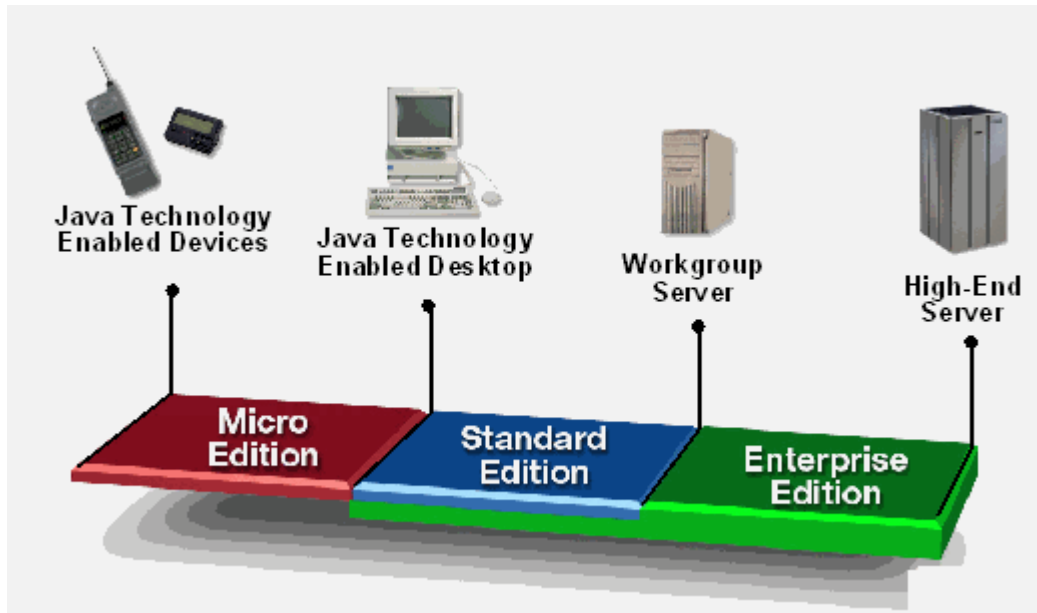


Figura 2: Ediciones de Java

### **2.1.2.- JAVA 2 PLATFORM, ESTÁNDAR EDITION (J2SE)**

Esta edición de Java es la más utilizada, y es la que recoge en cierta manera la iniciativa del lenguaje Java.

Contiene un conjunto de ejecución y de API's para crear aplicaciones no destinadas a ser centralizadas y de accesos concurrentes.

### **2.1.3.- JAVA 2 PLATFORM, ENTREPRISE EDITION (J2EE)**

Esta versión está orientada al entorno empresarial, ya que este entorno tiene unas características especiales: está pensado no para ser ejecutado en un equipo, sino para ejecutarse sobre una red de ordenadores de manera distribuida y remota.

Está especialmente enfocada al desarrollo de aplicaciones Web, ya que J2EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML ... y define cómo coordinarlos. J2EE también configura algunas especificaciones únicas para componentes. Estas incluyen Enterprise JavaBeans, servlets, Java Server Pages y varias tecnologías de servicios Web. Esto permite al



desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores.

Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

#### **2.1.4.- JAVA 2 PLATFORM, MICRO EDITION (J2ME)**

---

Esta versión de Java está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDAs o electrodomésticos inteligentes.

Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que sólo requiere unos pocos kilobytes de memoria para funcionar), la inclusión de un pequeño recolector de basura...

#### **2.1.5.- CONCLUSIONES**

---

Para el desarrollo de este proyecto, como es evidente, se ha utilizado Java en su edición Enterprise Edition, ya que es la que mejor se adaptaba a las necesidades de un proyecto Web, en el que se necesita compartir información a través de la red y además se puede utilizar en cualquier máquina que tenga instalada una máquina virtual de Java.

En un principio también se propuso como lenguaje PHP, pero aunque es un lenguaje mucho más sencillo que Java, suele utilizarse en pequeños proyectos y este proyecto puede tener futuras ampliaciones e ir creciendo con el tiempo, además respecto a ello, Java es más completo. Por otro lado, al usar Java, las páginas JSP y los servlet se compilan en memoria mientras que en PHP las páginas se compilan cada vez que son ejecutadas por el cliente, por lo que Java es mucho más rápido. Finalmente diversos estudios han comprobado que aunque ambas tecnologías son similares en cuanto a rendimiento para satisfacer una petición, PHP no es escalable, de manera que a medida que aumenta el número de usuarios, el rendimiento decrece y esta aplicación puede ser usada por muchísimos usuarios simultáneos.



## 2.2.- MySQL

### 2.2.1.- INTRODUCCIÓN

Los sitios Web modernos parecen depender cada vez más y más de sistemas de bases de datos complejos. Estos sistemas almacenan todos los datos críticos, y permiten un mantenimiento sencillo en la mayoría de los casos.

El Structured Query Language (SQL) es un lenguaje de base de datos muy popular y su estandarización hace bastante fácil almacenar, actualizar y procesar datos. Uno de los más poderosos servidores SQL es el llamado MySQL.



Figura 3: Logotipo de MySQL

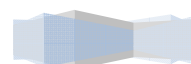
MySQL, es un sistema de gestión de bases de datos SQL de código abierto (Open Source) desarrollado, distribuido y soportado por la compañía MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Esta compañía Open Source de segunda generación une los valores y metodología Open Source con un exitoso modelo de negocio. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

SQL fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde ese entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003.

### 2.2.2.- UTILIZACIÓN DE MYSQL

Con MySQL, podemos gestionar bases de datos normales y distribuidas.

Gracias a este sistema podemos crear diferentes tablas mediante las cuales se podrá almacenar datos manteniéndolos de esta forma centralizados. Además permite la creación de disparadores ó triggers, la realización de consultas, la creación de vistas, de cursores, de procedimientos almacenados...



La centralización de la información supone un gran avance tanto en lo referente al almacenamiento y distribución de datos y, tanto o más, en cuanto a medidas de seguridad para evitar la pérdida de información (de esto se encarga una parte muy importante de las tecnologías de la información, que es la Auditoría Informática).

### 2.2.3.- TIPOS DE DATOS

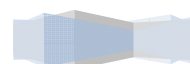
Después de la fase de diseño de una base de datos, y tras realizar el paso a tablas del mismo, es necesario crear las tablas correspondientes dentro de la base de datos. Para cada campo de cada una de las tablas, es necesario determinar el tipo de datos que contiene, para de esa forma ajustar el diseño de la base de datos, y conseguir un almacenamiento óptimo con la menor utilización de espacio.

Los tipos de datos que pueden existir en un campo, se pueden agrupar en tres grandes grupos:

➤ Tipos numéricos:

TIPO CAMPO	TAMAÑO ALMACENAMIENTO
<b>tinyint</b>	1 byte
<b>smallint</b>	2 bytes
<b>mediumint</b>	3 bytes
<b>int</b>	4 bytes
<b>integer</b>	4 bytes
<b>bigint</b>	8 bytes
<b>float(x)</b>	4 ó 8 bytes
<b>float</b>	4 bytes
<b>double</b>	8 bytes
<b>double precision</b>	8 bytes
<b>real</b>	8 bytes
<b>decimal(M,D)</b>	M+2 bytes si $D > 0$ , M+1 bytes si $D = 0$
<b>numeric(M,D)</b>	M+2 bytes if $D > 0$ , M+1 bytes if $D = 0$

Tabla 1: Datos numéricos MySQL



➤ Tipo fecha:

TIPO CAMPO	TAMAÑO ALMACENAMIENTO
<b>date</b>	3 bytes
<b>datetime</b>	8 bytes
<b>timestamp</b>	4 bytes
<b>time</b>	3 bytes
<b>year</b>	1 byte

Tabla 2: Datos tipo fecha MySQL

➤ Tipos cadena:

TIPO CAMPO	TAMAÑO ALMACENAMIENTO
<b>char(n)</b>	n bytes
<b>varchar(n)</b>	n +1 bytes
<b>tinyblob, tinytext</b>	Longitud+1 bytes
<b>blob, text</b>	Longitud +2 bytes
<b>mediumblob, mediumtext</b>	Longitud +3 bytes
<b>longblob, longtext</b>	Longitud +4 bytes
<b>enum('value1','value2',...)</b>	1 ó dos bytes dependiendo del número de valores
<b>set('value1','value2',...)</b>	1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores

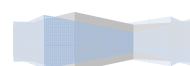
Tabla 3: Datos tipo cadena MySQL

## 2.2.4.- PRINCIPALES CARACTERÍSTICAS

Del sistema MySQL, podemos destacar que es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios, y la conectividad, velocidad y seguridad hacen que MySQL sea conveniente para acceder a bases de datos en Internet.

Otras de sus características son:

- Permite ser usado a través de clientes C, C++, Java, Perl y TLC.
- Es multiproceso, de forma que puede usar varias CPU a la vez.



- Puede trabajar en diferentes plataformas y sistemas operativos.
- Cuenta con un sistema de contraseñas y privilegios muy flexible y seguro.
- Utiliza registros de longitud fija y variable.
- ...

### 2.2.5.- CONCLUSIONES

---

MySQL es muy utilizado en aplicaciones Web y es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación.

En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones y por lo tanto para este Proyecto Fin de Carrera.

Otra posibilidad sería el uso de Oracle como gestor de BD, pero principalmente fue descartado porque es una BD propietaria y por lo tanto para poder utilizarla se debe incurrir en un coste.

## 2.3.- APACHE TOMCAT

---

### 2.3.1.- INTRODUCCIÓN

---

Tomcat es un contenedor de Servlet, que implementa las especificaciones de Servlet y Java Server Pages (JSP). Fue desarrollado bajo el proyecto de Jakarta en la Apache Software Foundation.



Figura 4: Logotipo de Tomcat

En sus comienzos existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.



Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la Máquina Virtual de Java.

### 2.3.2.- FICHEROS DE CONFIGURACIÓN

La configuración de Tomcat, se basa en dos ficheros de configuración:

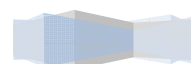
- **web.xml:** Configura los diferentes contextos de Tomcat. Este fichero se trata en la especificación del API Servlet, por lo que aquí omitiremos comentarlo.
- **server.xml:** Es el fichero de configuración principal de Tomcat y tiene dos objetivos. Por un lado se encarga de proporcionar la configuración inicial para los componentes de Tomcat y por otro lado se encarga de especificar su estructura, es decir, permite que Tomcat arranque y se construya por sí mismo. Los elementos más importantes de server.xml se describen en la siguiente tabla:

ELEMENTO	DESCRIPCIÓN
Server	Define un servidor Tomcat. Puede contener elementos Logger y ContextManager.
Logger	Define un objeto logger. Cada objeto de este tipo tiene un nombre que lo identifica, así como un path para el fichero log que contiene la salida.
ContextManager	Especifica la configuración y estructura para un conjunto de ContextInterceptor, Context y sus componentes.
ContextInterceptor y RequestInterceptor	Estos interceptores, escuchan ciertos eventos que suceden en el ContextManager, como por ejemplo, los eventos de arranque y parada de Tomcat.
Connector	Representa una conexión al usuario, a través de un servidor Web o directamente al navegador. Este objeto es responsable del control de hilos y de leer/escribir la peticiones/respuestas desde los sockets conectados a diferentes clientes.
Context	Cada context representa un path en el árbol de Tomcat donde situar nuestra aplicación Web. Contiene los siguientes elementos: El path donde se localiza el contexto. Puede ser completo ó relativo al home del ContextManager. Nivel de depuración usado para los mensajes de depuración. Una bandera reloadable que permite recargar Tomcat cuando hay un cambio de forma que se pruebe el nuevo código sin tener parar y arrancar el servidor.

Tabla 4: Elementos del fichero server.xml

### 2.3.3.- CONCLUSIONES

Todos estos elementos, y entre otros motivos su sencillez, tanto en el uso como en la configuración, hacen que Tomcat haya sido elegido como servidor Web para esta aplicación.





Además se destaca que al funcionar mediante un contenedor de Servlet a través de una máquina virtual, y teniendo en cuenta que como lenguaje de programación se utiliza Java, resulta más que recomendable su utilización.

Se comparó Apache Tomcat con el servidor Internet Information Services (IIS), pero se desechó su elección porque además de no ser un software de código abierto, lo cual encarecería el coste de la aplicación, tampoco es multiplataforma ya que solo puede utilizarse sobre Windows.

## 2.4.- STRUTS

### 2.4.1.- INTRODUCCIÓN

La Web ha tenido un largo trayecto, desde sus comienzos en 1984 hasta la actualidad, vamos a hablar un poco de ello.

La Web fue creada para intercambiar documentos estáticos aunque finalmente entraron en marcha contenidos dinámicos, gracias a la Common Gateway Interface (CGI). Aunque las aplicaciones CGI fueron muy buenas, contaban con una serie de limitaciones, lo que dio lugar a la creación de los Servlet de Java.

Los servlet proporcionaban un método independiente de la plataforma, basado en componentes para crear aplicaciones Web. Además eran más eficaces que los CGI pero su gran problema es que necesitaban una gran interacción con el navegador, por lo que finalmente se comenzaron a desarrollar aplicaciones Web basadas en Java Server Pages (JSP).

Tras todos estos sucesos y avances de la Web, los desarrolladores se dieron cuenta de que debían combinar las JSP y los servlets para desplegar aplicaciones Web. De esta forma aparecieron dos modelos:

- **Modelo 1:** Las páginas JSP gestionan todo el procesamiento de una petición y son las responsables de mostrar el resultado al cliente.

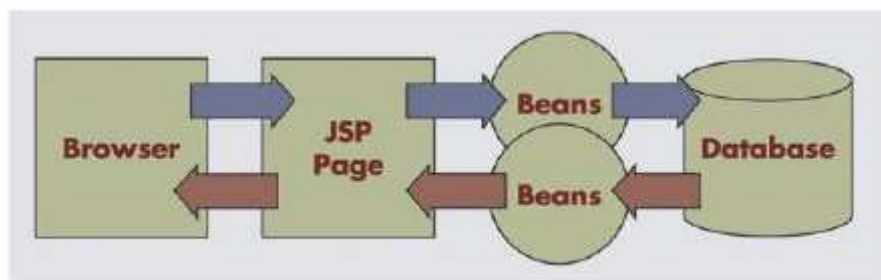
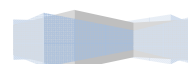


Figura 5: Modelo 1



- **Modelo 2:** La petición del cliente es interceptada por un servlet controlador que gestiona el procesamiento inicial de la petición y determina qué página JSP mostrar como respuesta a la petición.

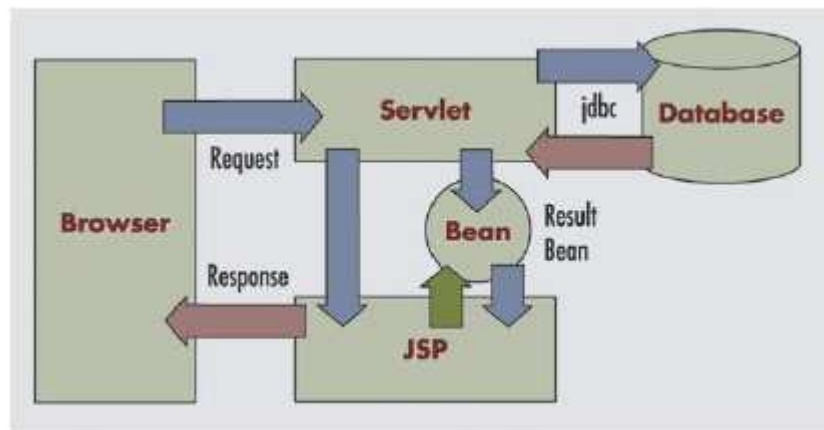


Figura 6: Modelo 2

El modelo 2 conocido comúnmente como modelo-vista-controlador (MVC) es un patrón de arquitectura software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos:

- El modelo, que es el responsable de la lógica de negocio.
- La vista, que es la responsable de presentar el contenido al cliente.
- El controlador, que se encarga de controlar el flujo y estado de la entrada de datos por parte del cliente y seleccionar la vista correspondiente.

El MVC funciona de la siguiente manera: el flujo de la aplicación está dirigido por un controlador central. El controlador delega solicitudes a un manejador apropiado. Los manejadores están unidos a un modelo, y cada manejador actúa como un adaptador entre la solicitud y el modelo. El modelo representa, o encapsula, un estado o lógica de negocio de la aplicación. Luego el control es devuelto a través del controlador hacia la vista apropiada. El reenvío puede determinarse consultando los conjuntos de mapeos, cargados desde un fichero de configuración en XML. Esto proporciona un acoplamiento cercano entre la vista y el modelo, que hace que las aplicaciones Web sean más fáciles de crear y de mantener.

El MVC dio lugar al nacimiento de Struts.

Struts es un marco de trabajo (framework) de código abierto creado por Craig R. McClanahan y donado a la Apache Software Foundation en el año 2000, que proporciona a los desarrolladores una infraestructura unificada sobre la que se pueden basar las aplicaciones Web. A través de este marco de trabajo, se permite que los desarrolladores se centren en la creación de la aplicación de negocio en lugar de las infraestructuras.



# Struts

Figura 7: Logotipo de Struts

## 2.4.2.- CONCLUSIONES

---

Dado que el marco de trabajo de Struts esta basado en la arquitectura de diseño MVC, es muy útil su utilización para esta aplicación, ya que MVC presenta las siguientes ventajas:

- Fácil de explicar y entender.
- Permite dividir el trabajo en base a distintos roles.
- Hay que tener en cuenta la mantenibilidad, las posibles variaciones futuras, como ampliaciones, mejoras o modificaciones de las funcionalidades ya existentes. Con esta arquitectura se reducen las dependencias y aumenta la modularidad, así como la potencia de reutilización.
- Por otro lado, es necesario que la arquitectura agrupe los componentes de tal manera que se consiga la máxima cohesión entre componentes de la misma capa y el mínimo acoplamiento entre capas distintas.
- Además, con esta arquitectura se consigue un acceso más eficiente a los datos, así como una gestión más eficiente de los componentes críticos de la aplicación.

Por otro lado, al ser parte del proyecto Jakarta, es compatible con las tecnologías comentadas anteriormente.

El marco de trabajo Struts, contiene diversos elementos, y varias características importantes de comentar, por ello, se adjunta una explicación mas extensa del marco, en *Anexo I.- El Marco de Trabajo Struts*.

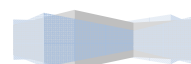
## 2.5.- HIBERNATE

---

### 2.5.1.- INTRODUCCIÓN

---

Para la mayoría de las aplicaciones, almacenar y recuperar información implica alguna forma de interacción con una base de datos relacional. Esto ha representado un problema fundamental para los desarrolladores ya que algunas veces el diseño de datos relacionales y los ejemplares orientados a objetos comparten estructuras de relaciones muy diferentes dentro de sus respectivos entornos.



Las bases de datos relacionales están estructuradas en una configuración tabular y los ejemplares orientados a objetos normalmente están relacionados en forma de árbol. Esta diferencia es lo que ha provocado que se desarrollen diversas tecnologías de persistencia de objetos intentando así construir un puente entre el mundo relacional y el mundo orientado a objetos.

Hibernate es un marco de trabajo que tiene por objetivo facilitar la persistencia de objetos Java en bases de datos relacionales y al mismo tiempo la consulta de estas bases de datos para obtener objetos, es decir, que Hibernate es una herramienta que realiza el mapeado entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM (object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.



Figura 8: Logotipo de Hibernate

Para lograr el mapeado de objetos relacionales a objetos Java, se permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL.

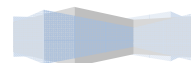
Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente.

## 2.5.2.- CONCLUSIONES

---

Es evidente la necesidad de la aplicación de almacenar datos en una base de datos relacional y por lo tanto se hace imprescindible su uso con el lenguaje Java, esto da lugar a que el uso de Hibernate sea necesario, ya que permitirá mapear ambos lenguajes y ahorrar mucho tiempo realizando consultas a través del HQL.

El marco de trabajo Hibernate, contiene otros elementos importantes que conviene comentar, por ello, se ha dedicado un anexo a ello, *Anexo II.- El Marco de Trabajo Hibernate*.



## 2.6.- JAVASERVER PAGES (JSP)

### 2.6.1.- INTRODUCCIÓN

Las páginas JSP, desarrolladas por Sun Microsystems, son un tipo especial de páginas que combinan HTML con fragmentos de Java para producir páginas Web dinámicas.

Las JSP permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas, a través de las librerías de etiquetas como TagLibs o incluso librerías personalizadas.

Su funcionamiento es el siguiente:

Cuando un cliente pide una página JSP del sitio Web y esta no se ha ejecutado antes, la página es inicialmente pasada al motor JSP, el cual se encarga de compilarla y convertirla en Servlet. Después la ejecuta y devuelve el contenido resultante al cliente en forma de página HTML.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java lo que permite separar en niveles o capas las aplicaciones Web, almacenando en clases Java las partes que consumen más recursos (así como las que requieren más seguridad) y dejando la parte encargada de formatear el documento HTML en el archivo JSP. La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información.

### 2.6.2.- SINTAXIS

Dentro de una página JSP podemos encontrarnos diferentes elementos de sintaxis como pueden ser:

- **Objetos implícitos:** Variables basadas en el API de Servlet que no necesitan ser declaradas. Los más importantes son:
  - *Request:* Instancia de `HttpServletRequest` que hace referencia a la petición del cliente.
  - *Response:* Instancia de `HttpServletResponse` que hace referencia a la respuesta del servidor.
  - *Session:* Instancia de `HttpSession` que permite acceder a la sesión para leer o escribir en ella.
  - *Out:* Instancia de `JspWriter` que permite escribir la respuesta.



- **Directivas:** Instrucciones para el contenedor, que serán tenidas en cuenta la hora de generar el servlet. Estas pueden ser:
  - *Page:* Permite especificar atributos relacionados con la página a procesar.
  - *Include:* Permite incluir el contenido de otro fichero dentro de la página.
  - *Taglib:* Permite declarar otras librerías de tag, para poder usarlas en la página.
- **Elementos de scripting:** Permiten la inserción de código Java dentro de una página. Estos elementos pueden ser de tres tipos:
  - *Declaraciones:* Se utilizan para definir variables y métodos con ámbito de clase para el servlet generado.
  - *Scriptlets:* Bloque de código Java ejecutado durante el procesamiento de la respuesta.
  - *Expresiones:* Se utiliza para declarar expresiones Java.
- **Acciones:** Acciones que llevan asociada cierta funcionalidad y que actúan en tiempo de ejecución. Son declaradas por medio de etiquetas XML.

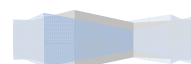
### 2.6.3.- LIBRERÍAS DE ETIQUETAS

Las librerías de etiquetas, se utilizan para simplificar el código y dar una mayor flexibilidad, de forma que se permita incluir lógica sin escribir directamente código Java.

Dentro de las librerías de etiquetas podemos destacar las propias de JSP y las JSTL. En la librería JSP podemos encontrar las siguientes etiquetas:

ETIQUETA	DESCRIPCIÓN
<b>jsp:include</b>	Permite insertar ficheros dentro de una página, en el momento en el que esta es solicitada.
<b>jsp:useBean</b>	Permite utilizar un JavaBean dentro de una página.
<b>jsp:setProperty</b>	Permite modificar el valor de una propiedad de un JavaBean instanciado anteriormente.
<b>jsp:getProperty</b>	Permite recuperar el valor de una propiedad de un JavaBean instanciado anteriormente.
<b>jsp:forward</b>	Permite reenviar una petición a otra página.
<b>Jsp:plugin</b>	Descarga un plugin de Java

Tabla 5: Etiquetas JSP



La librería JSTL es un componente dentro de la especificación de J2EE y es controlada por Sun Microsystems.

JSTL es un conjunto de librerías de etiquetas simples y estándares que permiten extender funcionalidades personalizadas por medio de TAGs incluidos dentro de las páginas JSP, invocando una determinada funcionalidad sin tener que escribir código Java en la página.

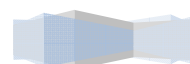
Las etiquetas JSTL se encuentran organizadas en 4 librerías:

- **Etiquetas Core:** Etiquetas para iteración sobre datos, operaciones condicionales, e importación de datos de otras páginas.

ETIQUETA	DESCRIPCIÓN
<b>c:out</b>	Muestra el resultado de una expresión. Su funcionalidad es equivalente a la de <code>&lt;%= %&gt;</code> .
<b>c:set</b>	Introduce/Modifica un valor en una variable.
<b>c:remove</b>	Elimina una variable.
<b>c:if</b>	Evalúa si una condición se cumple.
<b>c:choose,</b> <b>c:when,</b> <b>c:otherwise</b>	Evalúa si una condición se cumple y sino se cumple.
<b>c:forEach</b>	Permite iterar sobre los elementos siguientes: Arrays de objetos o tipos primitivos. Instancias de <code>java.util.Collection</code> , <code>Java.util.Map</code> , <code>java.util.Iterator</code> , <code>java.util.Enumeration</code> . Cadenas delimitadas por comas. Instancias de <code>javax.servlet.jsp.jstl.sql.Result</code> .
<b>c:forTokens</b>	Permite descomponer una cadena en tokens.
<b>c:import</b>	Proporciona toda la funcionalidad de <code>jsp:include</code> .
<b>c:param</b>	<code>c:import</code> puede tener hijos <code>c:param</code> , que se usan para pasar parámetros a la URL a recuperar.
<b>c:redirect</b>	Ejecuta un forward.
<b>c:url</b>	Sirve para mostrar una URL o para grabarla en una variable.
<b>c:catch</b>	Permite capturar y tratar los errores que se produzcan en una página.

**Tabla 6: Etiquetas Core de JSTL**

- **Etiquetas Xml:** La librería XML de JSTL usa XPath para obtener partes de los documentos XML con que opera, es decir, comprende el procesamiento de XML.
- **Etiquetas Fmt:** Comprende la internacionalización y formato de valores como de moneda y fechas.



ETIQUETA	DESCRIPCIÓN
<b>fmt:bundle</b>	Permite cargar recursos de mensajes específicos solo para una zona de la página
<b>fmt:message</b>	Acepta una clave y devuelve el resultado correspondiente a la clave en el fichero de recursos especificado.
<b>fmt:param</b>	Algunos mensajes aceptan parámetros, de forma parecida a los PreparedStatement de SQL.
<b>fmt:setBundle</b>	Permite cargar recursos de mensajes específicos.
<b>fmt:setLocale</b>	Permite establecer el valor de la Locale.
<b>fmt:formatNumber</b>	Permite dar formato a un valor numérico.
<b>fmt:formatDate</b>	Permite dar formato a una fecha.
<b>fmt:parseNumber</b>	Interpreta cadenas como números. Es necesario para pasar valores numéricos a algunas etiquetas como sql:param, fmt:formatNumber, y otras.
<b>fmt:parseDate</b>	Interpreta cadenas como fechas. Usa el locale del contenedor JSP.
<b>fmt:setTimeZone</b>	Permite cambiar la zona horaria.
<b>fmt:timeZone</b>	Permite cambiar temporalmente la zona horaria.

**Tabla 7: Etiquetas Fmt de JSTL**

- **Etiquetas Sql:** Comprende el acceso a base de datos.

## 2.6.4.- CONCLUSIONES

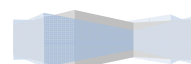
Debido al uso del lenguaje de programación Java, será necesario que las páginas Web asociadas a este proyecto sean también definidas en Java, por lo tanto la mejor opción es utilizar JSP.

Además destacamos que al hacer uso de Struts, este proporciona una importante librería de etiquetas a través del uso de JSP con lo cual es conveniente utilizar JSP con esta librería con el fin de simplificar el código en las páginas, fomentar la modularidad y reutilización y por supuesto, facilitar el posterior mantenimiento de las páginas.

## 2.7- JAVASCRIPT

### 2.7.1.- INTRODUCCIÓN

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.





Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores Web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995.

Este lenguaje consta de tres partes principales:

- **Client-side JavaScript:** La parte cliente de JavaScript permite la creación de aplicaciones que se ejecutan en un navegador.
- **Server-side JavaScript:** La parte servidor de JavaScript permite la creación de aplicaciones que se ejecutan en un servidor Web.
- **Core JavaScript:** La parte central de JavaScript contiene un conjunto de objetos básicos (Array, Date, Math...) además de los elementos del lenguaje como son los operadores, las estructuras de control y las sentencias.

## 2.7.2.- CONCLUSIONES

---

Utilizando JavaScript se pueden crear páginas HTML de forma dinámica que procesan las entradas del usuario y mantienen datos persistentes utilizando objetos especiales, ficheros y bases de datos relacionales. Además ofrece la posibilidad de comunicarse con aplicaciones escritas en Java, de modo que se puedan instanciar objetos Java desde el código JavaScript y también se pueda acceder a los objetos, propiedades y métodos JavaScript desde código Java. También permite la validación de datos sin necesidad de acceso a un servidor.

## 2.8- ECLIPSE WEB TOOLS PLATFORM (WTP)

---

### 2.8.1.- INTRODUCCIÓN

---

Eclipse es una plataforma de software de código abierto para desarrollar en Java, creada originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.





**Figura 9: Logotipo de Eclipse**

La definición que da el proyecto Eclipse acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".

Esta plataforma es considerada un Entorno de Desarrollo Integrado (IDE), ya que con ella es posible escribir código Java, compilarlo y ejecutarlo sin tener que cambiar de aplicación.

Eclipse emplea módulos (plug-in) para proporcionar toda su funcionalidad al frente de la plataforma, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software.

En cuanto a las aplicaciones clientes, Eclipse provee al programador con marcos de trabajo (frameworks) muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones Web, etc.

### **2.8.2.- CARACTERÍSTICAS**

---

La versión actual de Eclipse dispone de las siguientes características:

- Editor de texto.
- Resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias con JUnit, que es un conjunto de librerías creadas por Erich Gamma y Kent Beck utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.
- Control de versiones con Concurrent Versions System (CVS), que es una aplicación informática que mantiene el registro de todo el trabajo y los



cambios en los ficheros (código fuente principalmente) que forman un proyecto permitiendo que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.

- Integración con Apache Ant, que es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción.
- Asistentes para creación de proyectos, clases, tests, etc.
- Refactorización, que es una técnica de ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

### 2.8.3.- WTP

---

WTP es un plugin de Eclipse que proporciona API's específicas para J2EE centrados en el desarrollo de aplicaciones Web.



Figura 10: Logotipo de Eclipse WTP

Los principios clave sobre los que el proyecto Eclipse WTP funciona, son los siguientes:

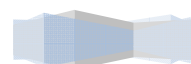
- **Extensión del valor de propuesta de Eclipse:** El Proyecto Eclipse ha establecido un alto estándar de excelencia técnica, innovación funcional y una extensibilidad general dentro del dominio de Java IDE. Intenta aplicar estos mismos estándares al dominio de herramientas Web/J2EE.
- **Ecosistema del vendedor:** El mayor objetivo de este proyecto es dar soporte a una herramienta vital de desarrollo de aplicaciones del mercado. Su base ejemplar será útil por sí misma pero será diseñada en principio para ser extensible; así los vendedores comerciales pueden usar los resultados generados por este proyecto para su propia innovación de producto y eficiencia de desarrollo.
- **Neutralidad del vendedor:** La neutralidad del vendedor estará situada en el núcleo de este proyecto. Se intenta motivar la participación en Eclipse y conducir su aceptación en el mercado mediante el fortalecimiento de las valiosas propuestas, a largo plazo del producto, del más amplio sector de vendedores de desarrollo de aplicaciones.



- **Innovación basada en estándares:** Este proyecto generará una base de herramientas basadas en estándares extensibles en el cual un amplio sector de vendedores podrá crear valiosos productos de desarrollo para sus clientes y usuarios. Donde quiera que exista un estándar, nos ceñiremos a él. Al menos, en principio, donde emerjan estándares, esperaremos a que emerjan; esto puede ser reevaluado más tarde de acuerdo a las necesidades del usuario y disponibilidad del colaborador.
- **Desarrollo ágil:** El objetivo es incorporar en el proceso de planificación las innovaciones que aparezcan una vez que un proyecto está en marcha y conseguir el apoyo de la comunidad de usuarios en lo que se logre hasta la fecha. Se piensa en un proceso de desarrollo y planificación ágil, en el cual en progreso sea incremental.
- **Diversidad y gama:** Pretenden asimilar las mejores ideas del gran número de participantes que representan las necesidades del enorme sector de usuarios finales. Por eso animan a las organizaciones a lo largo del sector técnico, mercado y dominios geográficos a participar en este proyecto.

En cuanto al alcance de este proyecto, al ser tan amplio, se podría decir que abarca todos los aspectos del desarrollo de software, pero realmente se concentra inicialmente en la infraestructura para herramientas utilizadas en la construcción de aplicaciones para entornos de ejecución Web y Java basados en estándares.

Fuera del alcance del proyecto, al menos en principio, existirá un soporte específico para arquitecturas de aplicación específicas para el vendedor, como ASP.Net y ColdFusion, o para extensiones no respaldadas por el JCP, como Apache Struts. Esto puede ser reevaluado más adelante de acuerdo a las necesidades del usuario y disponibilidad del colaborador y podría conducir a la creación de un nuevo subproyecto.



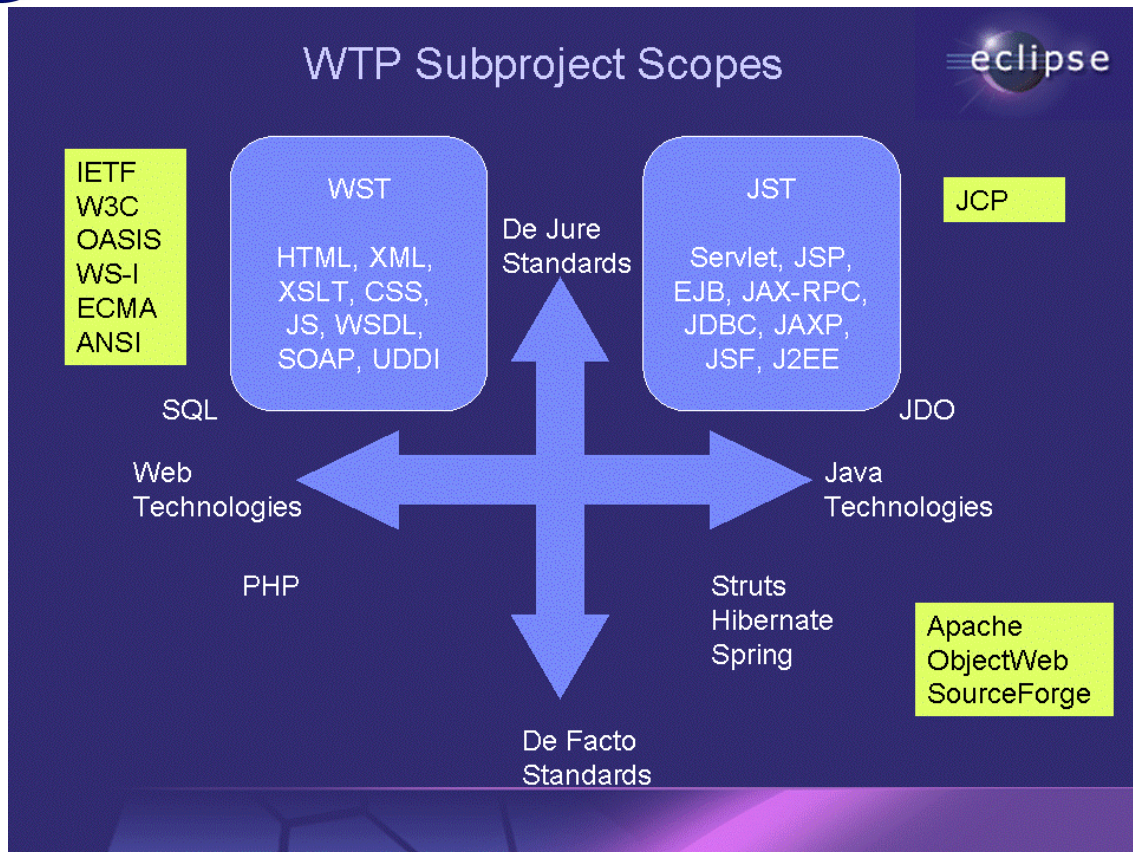


Figura 11: Alcance de los subproyectos de WTP

El proyecto esta limitado a proveer una infraestructura para el uso de herramientas en sí, en contraste con la infraestructura referente a la aplicación en tiempo de ejecución.

El objetivo final del proyecto es dar soporte a el uso de herramientas que permitan a los desarrolladores producir aplicaciones que otorguen un amplio grado de extensibilidad y reutilización junto con una eficiencia de desarrollo creciente.

## 2.8.4.- CONCLUSIONES

Para la realización de este Proyecto Fin de Carrera se ha optado por usar Eclipse WTP, ya que contiene ya integrados todos los elementos necesarios para crear una aplicación Web partiendo desde cero.



## 2.9- OTRAS HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

Además de todo lo comentado anteriormente, también se ha hecho uso de las siguientes herramientas y tecnologías:

- **Ubuntu:** Sistema operativo utilizado para desarrollar el proyecto y probar el servidor de aplicaciones.
- **Microsoft Office XP:** Para redactar toda la documentación del proyecto.
- **Microsoft Project 98:** Para realizar la planificación del proyecto.
- **SmartDraw 7:** Programa de modelado, a través del cual se han realizado todos los diagramas UML.
- **Internet Explorer y Mozilla Firefox:** Para probar la aplicación Web, descargar aplicaciones y buscar ayuda.
- **UML:** Lenguaje de modelado seguido para el diseño.





## 3.- ANÁLISIS DEL SISTEMA

Durante la etapa de análisis, se intenta definir correctamente el problema a resolver, obteniendo por medio de reuniones con el cliente y los diferentes miembros del equipo de desarrollo los requisitos del problema.

En primer lugar, se realizará la especificación de requisitos de usuario mediante la cual se pretende reunir un conjunto de requisitos que expresen y reflejen las necesidades y deseos, planteados por el cliente, de forma que estas queden formuladas de forma clara, completa y detallada.

Tras ello, se realizará un refinamiento de dichos requisitos, del cual surgirá el conjunto completo de requisitos del software que se desarrollará. Ese refinamiento se realizará aumentando el nivel de detalle en la especificación de cada requisito, de forma que, en conjunto, representen una solución clara a las necesidades reales del cliente, y eviten a su vez, que los diseñadores y programadores tengan que tomar decisiones relacionadas con dichas necesidades.

En consecuencia, el conjunto de requisitos software resultante constituirá la base para el diseño e implementación del sistema.

### 3.1.- REQUISITOS DE USUARIO

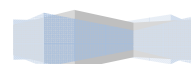
Los requisitos de usuario se dividen en dos grupos principales según su naturaleza:

- **Requisitos de capacidad:** Recogen todas las funcionalidades y características que debe contener la futura aplicación. Concretamente describen operaciones o secuencias de operaciones que el software debe ser capaz de realizar.
- **Requisitos de restricción:** Definen las restricciones impuestas por los usuarios sobre la manera en que el software resuelve los requisitos de capacidad.

El formato de cada uno de los requisitos es el siguiente:

IDENTIFICADOR			
Prioridad		Fuente	
Necesidad		Estabilidad	
Descripción			

Tabla 8: Formato de los requisitos de usuario



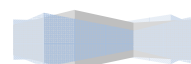
A continuación se detallan los campos que los componen:

- **Identificador:** Código único que identifica el requisito. Para los requisitos de capacidad el formato será RUC-XX, y para los requisitos de restricción el formato será RUR-XX, donde XX es un código ascendente de dos dígitos.
- **Prioridad:** Determina la prioridad del requisito respecto a otros. Su valor puede ser: alta, media o baja.
- **Fuente:** Determina el origen del requisito, que en la mayoría de los casos será el cliente.
- **Necesidad:** determina la importancia que tiene dentro del sistema el cumplimiento del requisito. Su valor puede ser: esencial, opcional o deseable.
- **Estabilidad:** Define si el requisito es prescindible o si es posible que en un futuro pueda ser modificado o incluso eliminado de la aplicación. Su valor puede ser: alta, media o baja.
- **Descripción:** Nombre y breve descripción textual del requisito.

### 3.1.1.- REQUISITOS DE CAPACIDAD

RUC-01			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Acceso al sistema:</i> El administrador, así como el coordinador y los profesores, deben acceder al sistema mediante la introducción de su identificador y su contraseña.		

RUC-02			
<b>Prioridad</b>	Media	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Desconexión del sistema:</i> El administrador, así como el coordinador, los profesores y los grupos de prácticas, pueden cerrar su sesión en el sistema.		

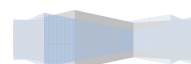




RUC-03			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Modificar contraseña:</i></p> <p>El administrador, los coordinadores, los profesores y los grupos de prácticas, podrán modificar su contraseña siempre que lo deseen.</p>		

RUC-04			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Gestionar personal docente:</i></p> <p>El administrador se encargará de gestionar todo el personal docente de la universidad que desee hacer uso de la aplicación. Esto implica que podrá:</p> <ul style="list-style-type: none"> <li>- Activar a los docentes para que puedan hacer uso de la aplicación.</li> <li>- Modificar los datos personales de los docentes que ha activado.</li> <li>- Consultar todos los docentes activados, ya sea realizando una búsqueda parametrizada u obteniendo todos los docentes.</li> </ul>		

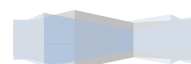
RUC-05			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Gestionar coordinadores:</i></p> <p>El administrador se encargará de gestionar todos los coordinadores de la universidad que deseen hacer uso de la aplicación. Esto implica que podrá:</p> <ul style="list-style-type: none"> <li>- Activar coordinadores para una asignatura en un curso académico.</li> <li>- Modificar los datos personales de los coordinadores que ha activado.</li> <li>- Consultar todos los coordinadores existentes en el sistema, ya sea realizando una búsqueda parametrizada u obteniendo todos los coordinadores.</li> <li>- Eliminar coordinadores del sistema.</li> </ul>		



RUC-06			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Gestionar profesores:</i></p> <p>El coordinador de cada asignatura, podrá gestionar los profesores de dicha asignatura, por lo que podrá realizar las siguientes acciones:</p> <ul style="list-style-type: none"> <li>- Activar profesores para la asignatura coordinada.</li> <li>- Añadir a dichos profesores privilegios para que puedan gestionar prácticas.</li> <li>- Modificar los datos personales de los profesores que ha activado.</li> <li>- Consultar todos los profesores existentes en el sistema, ya sea realizando una búsqueda parametrizada u obteniendo todos los profesores.</li> <li>- Eliminar profesores del sistema.</li> </ul>		

RUC-07			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Gestionar prácticas:</i></p> <p>Tanto los coordinadores de una asignatura como los profesores con permisos, podrán gestionar las prácticas de una asignatura, esto es:</p> <ul style="list-style-type: none"> <li>- Crear prácticas introduciendo su nombre y todos los parámetros necesarios para su configuración (listas de alumnos, corrector, archivos...).</li> <li>- Modificar las prácticas que están dadas de alta.</li> <li>- Consultar las prácticas existentes, ya sea realizando una búsqueda parametrizada u obteniendo todos los prácticas.</li> <li>- Eliminar las prácticas del sistema.</li> </ul> <p>Además, los coordinadores y profesores, podrán obtener la URL de una práctica y recoger las últimas prácticas que han sido entregadas por los grupos de prácticas.</p>		

RUC-08			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Gestionar alumnos:</i></p> <p>Coordinadores y profesores podrán importar, modificar, borrar y consultar listados de todos los alumnos matriculados en una asignatura en un curso académico. Además, podrán consultar los grupos que se han dado de alta para realizar las prácticas y ver todas las entregas.</p>		



RUC-09			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Verificar firma digital:</i> Coordinadores y profesores podrán verificar si la firma del archivo entregado por un grupo de práctica, se corresponde con la clave pública del servidor, con el fin de evitar que los alumnos reclamen sin razón alguna.		

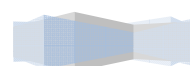
RUC-10			
<b>Prioridad</b>	Media	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<i>Modificar asignatura:</i> Los coordinadores y/o profesores que lo son de varias asignaturas, pueden alternar entre unas y otras, sin necesidad de cerrar sesión y volver a identificarse.		

RUC-11			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Gestionar grupo de autores:</i> Los alumnos de una asignatura podrán crear y modificar su grupo de prácticas.		

RUC-12			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Entregar prácticas:</i> Los alumnos podrán entregar electrónicamente a través de la aplicación los ficheros de la práctica para su posterior corrección por los profesores de la asignatura.		

### 3.1.2.- REQUISITOS DE RESTRICCIÓN

RUR-01			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Entorno amigable:</i> La interfaz gráfica de la aplicación debe ser de fácil manejo e intuitiva, ya que será utilizada por usuarios que pueden o no contar con conocimientos informáticos.		



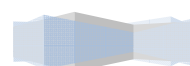
RUR-02			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Entorno seguro:</i></p> <p>El administrador, los coordinadores y todos los profesores deben trabajar en un entorno seguro. Será necesario que antes de entrar en el sistema se identifiquen introduciendo su identificador y contraseña.</p>		

RUR-03			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Protocolo de seguridad:</i></p> <p>Todas las páginas Web de la aplicación funcionarán sobre HTTPS, es decir, que toda la información que se envíe a través de ellas, se hará de forma cifrada, evitando así que posibles atacantes consigan información sensible, como contraseñas, prácticas...</p>		

RUR-04			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Entorno operativo:</i></p> <p>La aplicación será compatible con los navegadores Microsoft Internet Explorer 8 y Mozilla Firefox 3 o superior.</p>		

RUR-05			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Baja
<b>Descripción</b>	<p><i>Idioma:</i></p> <p>La interfaz gráfica del sistema se encontrara íntegramente en castellano.</p>		

RUR-06			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Contraseña:</i></p> <p>La contraseña será de al menos 7 caracteres permitiendo introducir caracteres de todo tipo (letras, números, signos de puntuación...) con la finalidad de hacerla mucho más segura.</p>		



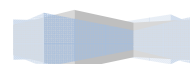
RUR-07			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Arquitectura del sistema:</i> El sistema seguirá una arquitectura en capas a través del patrón MVC (Modelo Vista Controlador) consiguiendo así una mayor reutilización y modularidad de la aplicación.		

RUR-08			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Sistema operativo:</i> La aplicación servidora, debe funcionar en entornos Unix y Windows.		

RUR-09			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Baja
<b>Descripción</b>	<i>Sistema de compresión:</i> Todos los ficheros que necesiten ser comprimidos, tendrán extensión .zip.		

RUR-10			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Baja
<b>Descripción</b>	<i>Formato de listados:</i> Todos los listados generados o importados, tendrán extensión .cvs, es decir, texto delimitado por puntos y comas.		

RUR-11			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Baja
<b>Descripción</b>	<i>Resolución de pantalla:</i> La aplicación se verá de forma correcta en pantallas configuradas a 1024*768 píxeles y también 1280*1024 píxeles.		



## 3.2.- REQUISITOS SOFTWARE

Los requisitos software describen principalmente lo que debe de hacer el sistema y tienen un alto grado de importancia en cuanto a verificación y seguimiento de la aplicación, ya que está, deberá cubrir todos y cada uno de los requisitos identificados para afirmar que está completa y el funcionamiento de los mismos se comprobará mediante el plan de pruebas.

Antes de comenzar a enumerar los requisitos es conveniente dar una definición de cada uno de los grupos en los que se engloban.

- **Requisitos funcionales:** Definen qué tiene que hacer el sistema.
- **Requisitos de interfaz:** Definen el modo en que interactúan y se comunican las interfaces y diferentes módulos del sistema.
- **Requisitos de recursos:** Define los recursos mínimos para que la aplicación pueda ponerse en marcha.
- **Requisitos de comprobación:** Define todas las comprobaciones que deberá realizar el sistema para que la interacción usuario-sistema sea la adecuada.
- **Requisitos de documentación:** Define las características que debe cumplir la documentación de la aplicación.
- **Requisitos de seguridad:** Define las características que debe cumplir la aplicación para ser segura.

El formato de estos requisitos será igual que el mencionado en el apartado 3.1.- *Requisitos de Usuario*, por lo que únicamente haremos mención del campo “Identificación”, que en este caso se compondrá de un acrónimo que identifica el tipo de requisito seguido de un guión y un número irreplicable dentro del mismo tipo de requisito: Los acrónimos son los siguientes:

- **RSF:** Requisitos software funcionales.
- **RSI:** Requisitos software de interfaz.
- **RSR:** Requisitos software de recursos.
- **RSC:** Requisitos software de comprobación.
- **RSD:** Requisitos software de documentación.
- **RSS:** Requisitos software de seguridad.



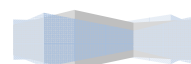
### 3.2.1.- REQUISITOS SOFTWARE FUNCIONALES

RSF-01			
<b>Prioridad</b>	Alta	<b>Fuente</b>	
<b>Necesidad</b>		<b>Estabilidad</b>	
<b>Descripción</b>	<p><i>Identificarse:</i></p> <p>Para acceder al sistema, al administrador, los coordinadores y los profesores deben identificarse introduciendo su identificador y contraseña. Además los coordinadores y los profesores, al poder serlo de varias asignaturas, deberán seleccionar la asignatura a la que desean acceder.</p>		

RSF-02			
<b>Prioridad</b>	Media	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Cerrar sesión:</i></p> <p>Cualquier usuario que se haya identificado en el sistema, podrá cerrar su sesión como usuario.</p>		

RSF-03			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Cambiar contraseña:</i></p> <p>Cualquier usuario que se haya identificado en el sistema, tiene la posibilidad de cambiar su contraseña de acceso. Para cambiar la contraseña el usuario debe introducir su antigua contraseña y dos veces la nueva, a excepción de los grupos de prácticas, que solo deben introducir dos veces la nueva contraseña.</p>		

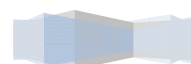
RSF-04			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Activar docente:</i></p> <p>El administrador podrá activar nuevos docentes, para ello deberá introducir los siguientes datos:</p> <ul style="list-style-type: none"> <li>- Identificador de usuario.</li> <li>- Contraseña.</li> <li>- Repetición de la contraseña.</li> <li>- NIA.</li> <li>- Nombre.</li> <li>- Apellidos.</li> <li>- E-mail.</li> </ul>		



RSF-05			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Modificar docente:</i></p> <p>Tras activar un docente o tras realizar una búsqueda, el administrador podrá modificar cualquiera de los datos del coordinador excepto su identificador y contraseña.</p>		

RSF-06			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Consultar docente:</i></p> <p>El administrador podrá consultar los datos de los docentes con el fin de visualizarlos, modificarlos o incluso borrarlos. Para ello contará con dos formas diferentes de realizar la consulta:</p> <ul style="list-style-type: none"> <li>- Búsqueda parametrizada, en la que el administrador deberá introducir uno o varios de los parámetros de búsqueda, que son: <ul style="list-style-type: none"> <li>o NIA.</li> <li>o Nombre.</li> <li>o Apellidos.</li> </ul> </li> <li>- Búsqueda de todos los docentes existentes en el sistema.</li> </ul>		

RSF-07			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Activar coordinador:</i></p> <p>El administrador podrá activar nuevos coordinadores, para ello deberá introducir los siguientes datos:</p> <ul style="list-style-type: none"> <li>- Seleccionar en un campo desplegable el nombre del docente que desea activar como coordinador.</li> <li>- Seleccionar en un campo desplegable la titulación y el plan de estudios a la que pertenece la asignatura.</li> <li>- Seleccionar en un campo desplegable la asignatura y el campus donde se encuentra.</li> <li>- Año académico.</li> </ul>		

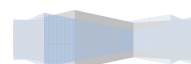




RSF-08			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Modificar coordinador:</i></p> <p>Tras activar un coordinador o tras realizar una búsqueda, el administrador podrá modificar cualquiera de los datos del coordinador, a excepción del nombre del coordinador.</p>		

RSF-09			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Consultar coordinador:</i></p> <p>El administrador podrá consultar los datos de los coordinadores con el fin de visualizarlos, modificarlos o incluso borrarlos. Para ello contará con dos formas diferentes de realizar la consulta:</p> <ul style="list-style-type: none"> <li>- Búsqueda parametrizada, en la que el administrador deberá introducir uno o varios de los parámetros de búsqueda, que son: <ul style="list-style-type: none"> <li>o NIA.</li> <li>o Nombre.</li> <li>o Apellidos.</li> <li>o Año académico.</li> <li>o Seleccionar en un campo desplegable la titulación y el plan de estudios.</li> <li>o Seleccionar en un campo desplegable la asignatura y el campus.</li> </ul> </li> <li>- Búsqueda de todos los coordinadores existentes en el sistema.</li> </ul>		

RSF-10			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Borrar coordinador:</i></p> <p>Tras activar un coordinador o tras realizar una búsqueda, el administrador podrá, si así lo desea, borrarlo.</p>		

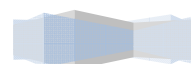


RSF-11			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Activar profesor:</i></p> <p>El coordinador de cada asignatura podrá activar nuevos profesores, para ello deberá introducir los siguientes datos:</p> <ul style="list-style-type: none"> <li>- Seleccionar en un campo desplegable el nombre del docente que desea activar como profesor de la asignatura que coordina.</li> <li>- Indicar si tiene o no permisos para gestionar prácticas.</li> </ul>		

RSF-12			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Modificar profesor:</i></p> <p>Tras activar un profesor o tras realizar una búsqueda, el coordinador podrá modificar si el profesor tiene o no permisos sobre las prácticas.</p>		

RSF-13			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Consultar profesor:</i></p> <p>El coordinador podrá consultar los datos de los profesores con el fin de visualizarlos, modificarlos o incluso borrarlos. Para ello contará con dos formas diferentes de realizar la consulta:</p> <ul style="list-style-type: none"> <li>- Búsqueda parametrizada, en la que el coordinador deberá introducir uno o varios de los parámetros de búsqueda, que son: <ul style="list-style-type: none"> <li>o NIA.</li> <li>o Nombre.</li> <li>o Apellidos.</li> </ul> </li> <li>- Búsqueda de todos los profesores existentes en el sistema que pertenezcan a la asignatura del coordinador.</li> </ul>		

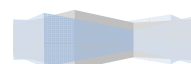
RSF-14			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Borrar profesor:</i></p> <p>Tras activar un profesor o tras realizar una búsqueda, el coordinador podrá, si así lo desea, borrarlo.</p>		



RSF-15			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Crear práctica:</i></p> <p>El coordinador o los profesores con permisos podrán dar de alta nuevas prácticas, para ello deberá introducir los siguientes datos:</p> <ul style="list-style-type: none"> <li>- Nombre de la práctica.</li> <li>- Indicar la convocatoria.</li> <li>- Seleccionar a través de un calendario la fecha tope de entrega.</li> <li>- Hora máxima de entrega.</li> <li>- Seleccionar en un campo desplegable el responsable de la práctica.</li> <li>- Añadir los datos sobre los autores: <ul style="list-style-type: none"> <li>o Número de autores.</li> <li>o Indicar si deben estar matriculados.</li> <li>o Indicar si se permiten grupos mixtos.</li> </ul> </li> <li>- Añadir los datos de los archivos entregables: <ul style="list-style-type: none"> <li>o Nombre.</li> <li>o Extensiones.</li> </ul> </li> </ul>		

RSF-16			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Modificar práctica:</i></p> <p>Tras dar de alta una práctica o tras realizar una búsqueda, se podrá modificar cualquiera de los datos de la práctica. Se permite incluso eliminar los archivos entregables.</p>		

RSF-17			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Consultar práctica:</i></p> <p>El coordinador o los profesores con permisos podrán consultar los datos de una práctica con el fin de visualizarla, modificarla o incluso borrarla. Para ello contará con dos formas diferentes de realizar la consulta:</p> <ul style="list-style-type: none"> <li>- Búsqueda parametrizada, en la que se deberá introducir uno o varios de los parámetros de búsqueda, que son: <ul style="list-style-type: none"> <li>o Nombre de la práctica.</li> <li>o Indicar convocatoria.</li> <li>o Fecha máxima de entrega.</li> <li>o Responsable.</li> </ul> </li> <li>- Búsqueda de todas las prácticas existentes en el sistema que pertenezcan a la asignatura del coordinador o profesor con permisos.</li> </ul>		

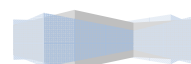


RSF-18			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Borrar práctica:</i> Tras dar de alta una práctica o tras realizar una búsqueda, el coordinador o profesor con permisos podrá, si así lo desea, borrarla.		

RSF-19			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Obtener URL:</i> Los coordinadores y profesores tras seleccionar en un campo desplegable el nombre de una práctica y la convocatoria, podrán obtener la URL de dicha práctica, para poder colocar el enlace de forma accesible para su futuro uso por los alumnos.		

RSF-20			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Recoger prácticas:</i> Los coordinadores y profesores tras seleccionar en un campo desplegable el nombre de una práctica y la convocatoria, podrán obtener un fichero con extensión .zip con la última versión de todas las prácticas entregadas por los grupos.		

RSF-21			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Importar listado:</i> Los coordinadores y profesores con permisos, podrán importar listados de alumnos para una asignatura en un curso académico. Para ello, deberán introducir los siguientes datos. <ul style="list-style-type: none"> <li>- Nombre.</li> <li>- Turno.</li> <li>- Indicar la ruta donde se encuentra el listado de extensión .cvs.</li> </ul>		

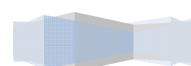


RSF-22			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Modificar listado:</i></p> <p>Tras dar de alta un listado o tras realizar una consulta, los coordinadores y profesores con permisos, podrán modificar cualquiera de los datos del listado.</p>		

RSF-23			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Consultar listado:</i></p> <p>Los coordinadores y profesores, podrán consultar los listados de alumnos de una asignatura en un curso académico concreto.</p>		

RSF-24			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Borrar listado:</i></p> <p>Tras dar de alta un listado o tras realizar una consulta, los coordinadores y profesores con permisos, podrán si así lo desean, borrarlo.</p>		

RSF-25			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Consultar grupos registrados:</i></p> <p>Los coordinadores y profesores podrán consultar los datos de los alumnos que se han registrado como grupo de prácticas. De cada grupo de prácticas, podrán visualizar:</p> <ul style="list-style-type: none"> <li>- Los datos del grupo y sus componentes.</li> <li>- Todas las entregas realizadas para cada práctica de la asignatura. Además tendrán la posibilidad de descargarlas.</li> </ul>		

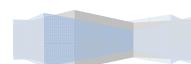


RSF-26			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Comprobar firma digital:</i></p> <p>A veces los alumnos reclaman en los despachos de los profesores, y para ello deben presentar la entrega de la práctica de la cual desean reclamar junto con la firma digital correspondiente generada por la aplicación. De este modo, los coordinadores y/o profesores deben introducir dicho fichero comprimido en .zip para que el sistema verifique si los datos se corresponden con la firma digital y la clave pública del servidor, con el fin de evitar engaños por parte de los alumnos.</p>		

RSF-27			
<b>Prioridad</b>	Media	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<p><i>Cambiar de asignatura:</i></p> <p>Tras pulsar sobre esta opción, los coordinadores y/o profesores visualizarán un listado con todas las asignaturas que coordinan e imparten, pudiendo seleccionar una de ellas, para acceder a la aplicación.</p>		

RSF-28			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Crear grupo de autores:</i></p> <p>Los alumnos podrán crear un grupo de prácticas, para ello deberán introducir los siguientes datos:</p> <ul style="list-style-type: none"> <li>- Nombre del grupo.</li> <li>- Contraseña.</li> <li>- Repetir contraseña.</li> <li>- Para cada alumnos que conforme el grupo deberán introducir: <ul style="list-style-type: none"> <li>o NIA.</li> <li>o Nombre.</li> <li>o Apellidos.</li> </ul> </li> </ul>		

RSF-29			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Modificar grupo de autores:</i></p> <p>Los alumnos tras dar de alta su grupo de prácticas o validarse en el sistema introduciendo el nombre del grupo y su contraseña, podrán modificar todos los datos del grupo a excepción del nombre del grupo.</p>		



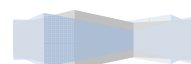
RSF-30			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Entrega de práctica:</i></p> <p>Los alumnos podrán realizar la entrega de prácticas electrónicamente tras darse de alta o validarse en el sistema introduciendo el nombre del grupo y su contraseña. Tras ello deberán seleccionar mediante un botón “Examinar” los ficheros a entregar.</p> <p>Después de la entrega, aparecerá una pantalla que les indicará si la entrega se ha realizado correctamente además de indicarles que deben descargar la práctica y el certificado, con el cual podrán realizar reclamaciones en caso de ser necesario.</p>		

### 3.2.2.- REQUISITOS SOFTWARE DE INTERFAZ

RSI-01			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Entorno amigable:</i></p> <p>La interfaz gráfica de la aplicación debe ser de fácil manejo e intuitiva, ya que será utilizada por usuarios que pueden o no contar con conocimientos informáticos.</p>		

RSI-02			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Baja
<b>Descripción</b>	<p><i>Idioma:</i></p> <p>La interfaz gráfica del sistema se encontrara íntegramente en castellano.</p>		

RSI-03			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<p><i>Botones volver:</i></p> <p>La interfaz gráfica contará, siempre que sea necesario, con botones para volver a la pantalla anterior, colocados en la parte inferior izquierda de la pantalla.</p>		



RSI-04			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Menús:</i> La interfaz gráfica mantendrá en su parte izquierda, un menú con todas las acciones posibles a realizar por un usuario.		

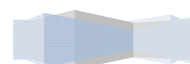
### 3.2.3.- REQUISITOS SOFTWARE DE RECURSOS

RSR-01			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Media
<b>Descripción</b>	<i>Protocolo de Internet:</i> Se utilizará HTTPS como protocolo de Internet, por lo que, será necesario que el usuario disponga de conexión a Internet para poder utilizar la aplicación.		

RSR-02			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Media
<b>Descripción</b>	<i>Software:</i> Se utilizará MySQL como Gestor de Bases de Datos como pasarela entre los almacenes de datos y la WEB.		

RSR-03			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Opcional	<b>Estabilidad</b>	Media
<b>Descripción</b>	<i>Entorno operativo:</i> La aplicación será compatible con los navegadores Microsoft Internet Explorer 8 y Mozilla Firefox 3 o superior.		

RSR-04			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Baja
<b>Descripción</b>	<i>Resolución de pantalla:</i> La aplicación se verá de forma correcta en pantallas configuradas a 1024*768 píxeles y también 1280*1024 píxeles.		





RSR-05			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Sistema operativo:</i> La aplicación servidora, debe funcionar en entornos Unix y Windows.		

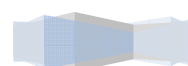
### 3.2.4.- REQUISITOS SOFTWARE DE COMPROBACIÓN

RSC-01			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Identificador de usuario:</i> Deberá comenzar por una letra y solo aceptará números o letras. La longitud mínima del campo será 5 caracteres.		

RSC-02			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	No esta permitido activar varios docentes, coordinadores o profesores con el mismo identificador de usuario.		

RSC-03			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>N/A:</i> Campo numérico cuya longitud debe ser 9 caracteres y debe comenzar por 100.		

RSC-04			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	No esta permitido dar de alta varios grupos de prácticas con los mismos alumnos para una misma asignatura.		



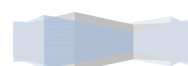
RSC-05			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Media
<b>Descripción</b>	<i>Contraseña:</i> Mínimo 7 caracteres, que podrán ser de todo tipo.		

RSC-06			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Repetir contraseña:</i> Comprobar que la contraseña repetida corresponde con la primera introducida.		

RSC-07			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Comprobar correo:</i> Comprobar que el correo electrónico introducido tiene el formato adecuado y que detrás de la @ se encuentra un nombre de dominio.		

RSC-08			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Media
<b>Descripción</b>	<i>Nombre del grupo:</i> Deberá comenzar por una letra y solo aceptará números o letras. No podrá contener " " _ .		

RSC-09			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	Comprobar que dos grupos para una misma asignatura no tienen el mismo nombre.		



RSC-10			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Ficheros:</i> Comprobar que los ficheros introducidos tienen el mismo nombre y extensión que los especificados.		

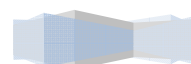
RSC-11			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Campos numéricos:</i> Comprobar que los valores de los campos numéricos son enteros y no exceden del rango predeterminado.		

RSC-12			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Campos obligatorios:</i> Comprobar que todos los campos obligatorios han sido rellenados.		

### 3.2.5.- REQUISITOS SOFTWARE DE DOCUMENTACIÓN

RSD-01			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Manual de usuario:</i> Se entregará junto al software final un manual de usuario completo y detallado, para cada uno de los perfiles que pueden utilizar la aplicación.		

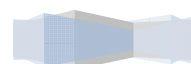
RSD-02			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Yankoa López
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Javadoc:</i> La documentación del software deberá ser generada a través de la herramienta Javadoc.		



### 3.2.6.- REQUISITOS SOFTWARE DE SEGURIDAD

RSS-01			
<b>Prioridad</b>	Alta	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Esencial	<b>Estabilidad</b>	Alta
<b>Descripción</b>	<i>Entorno seguro:</i> El administrador, los coordinadores y todos los profesores deben trabajar en un entorno seguro. Será necesario que antes de entrar en el sistema se identifiquen introduciendo su identificador y contraseña.		

RSS-02			
<b>Prioridad</b>	Media	<b>Fuente</b>	Cliente
<b>Necesidad</b>	Deseable	<b>Estabilidad</b>	Media
<b>Descripción</b>	<i>Protocolo de seguridad:</i> Todas las páginas Web de la aplicación funcionarán sobre HTTPS, es decir, que toda la información que se envíe a través de ellas, se hará de forma cifrada, evitando así que posibles atacantes consigan información sensible, como contraseñas, prácticas...		



## 4.- DISEÑO DEL SISTEMA

En este apartado se especifica la solución software propuesta para el diseño de la aplicación. En este caso, se reflejan los modelos y diagramas necesarios para indicar como debe construirse el sistema.

### 4.1.- CASOS DE USO

#### 4.1.1.- ESPECIFICACIÓN GRÁFICA DE LOS CASOS DE USO

Mediante el diagrama de casos de uso, se proporciona una visión de los requisitos funcionales del proyecto que especifican el funcionamiento del sistema, reflejándose la interacción entre los actores y el propio sistema.

En este diagrama, los casos de uso que son iguales pero usados por distintos actores, se han representado con el mismo nombre y un apóstrofe, por ejemplo el caso de uso “*Identificarse*” del administrador e “*Identificarse*” del coordinador y el profesor, son realmente el mismo caso de uso.

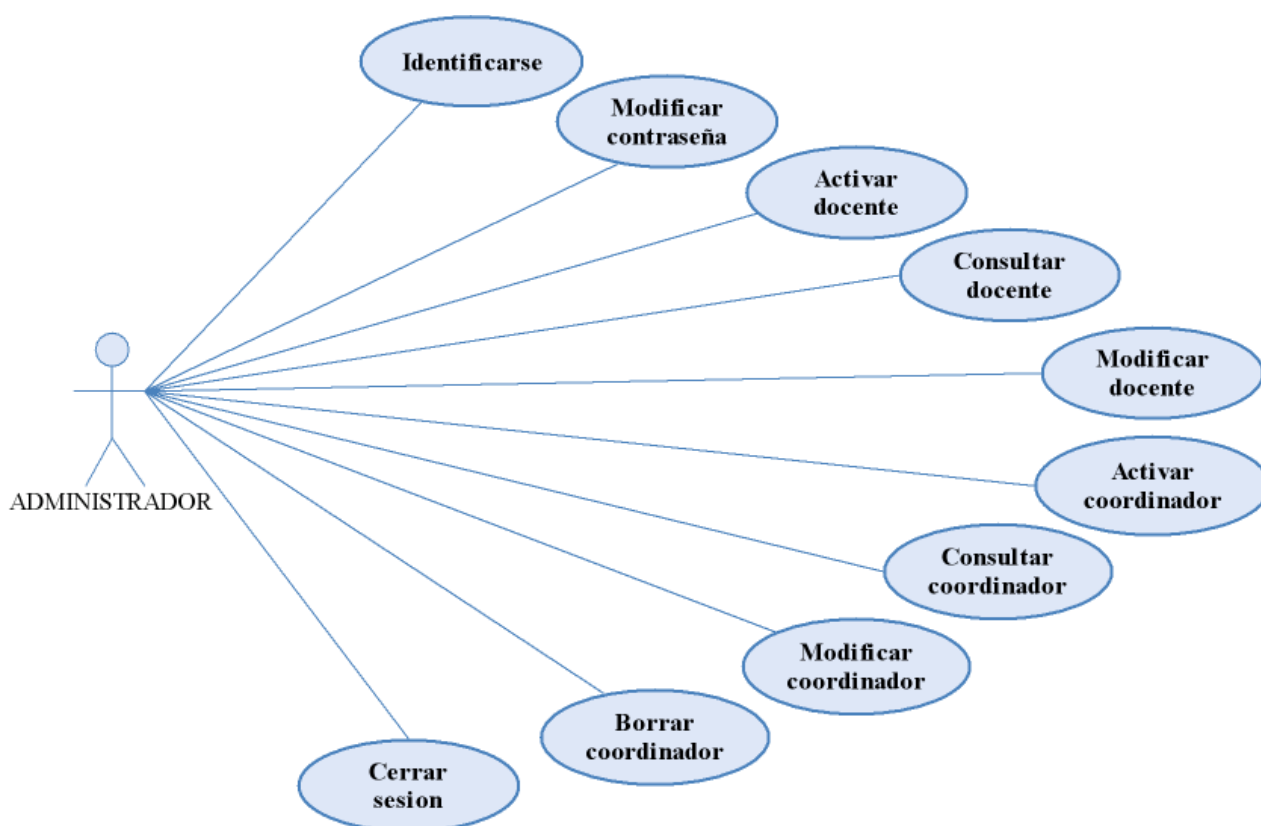
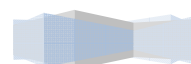


Figura 12: Casos de uso I



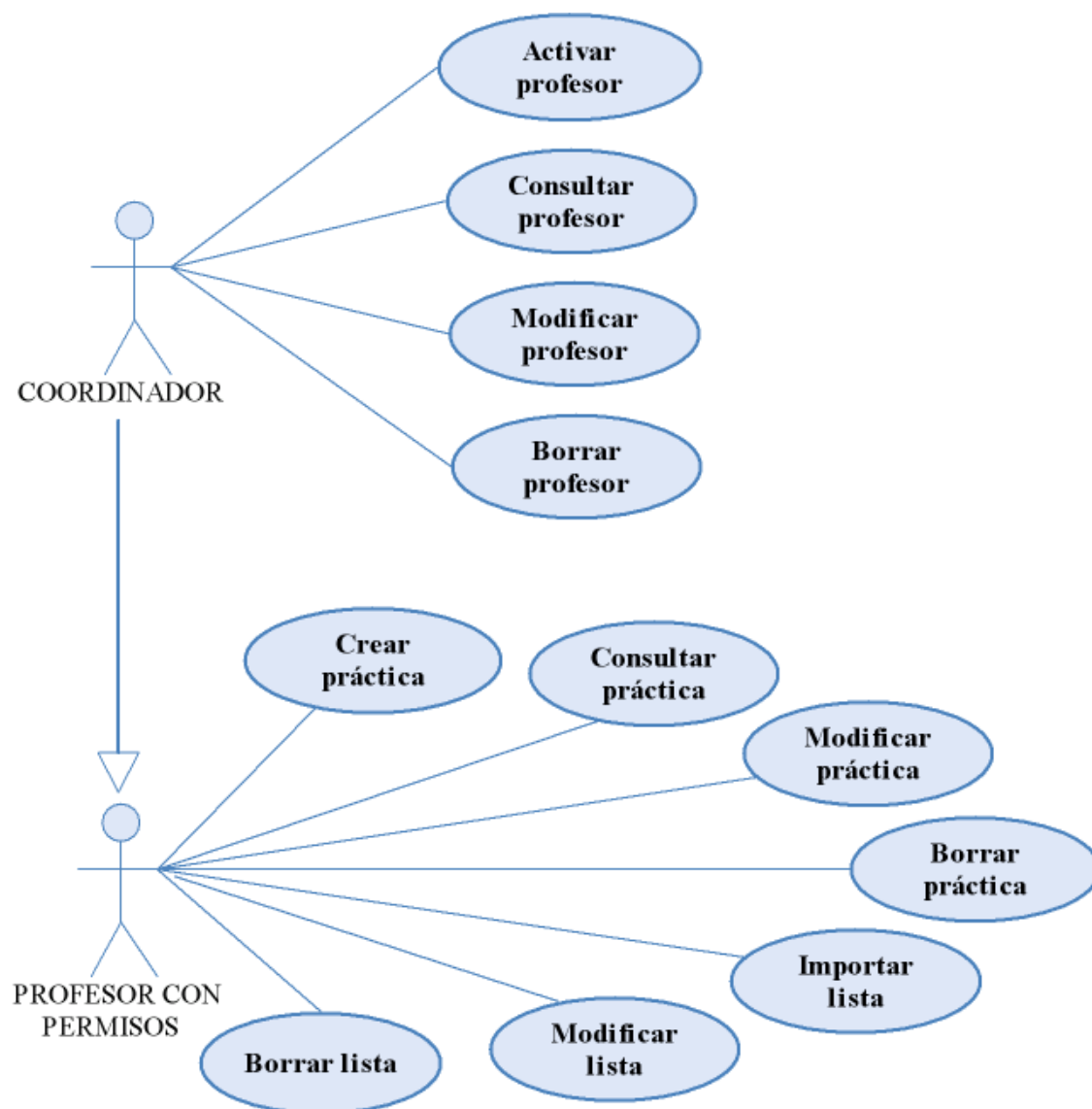
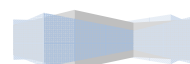


Figura 13: Casos de uso II



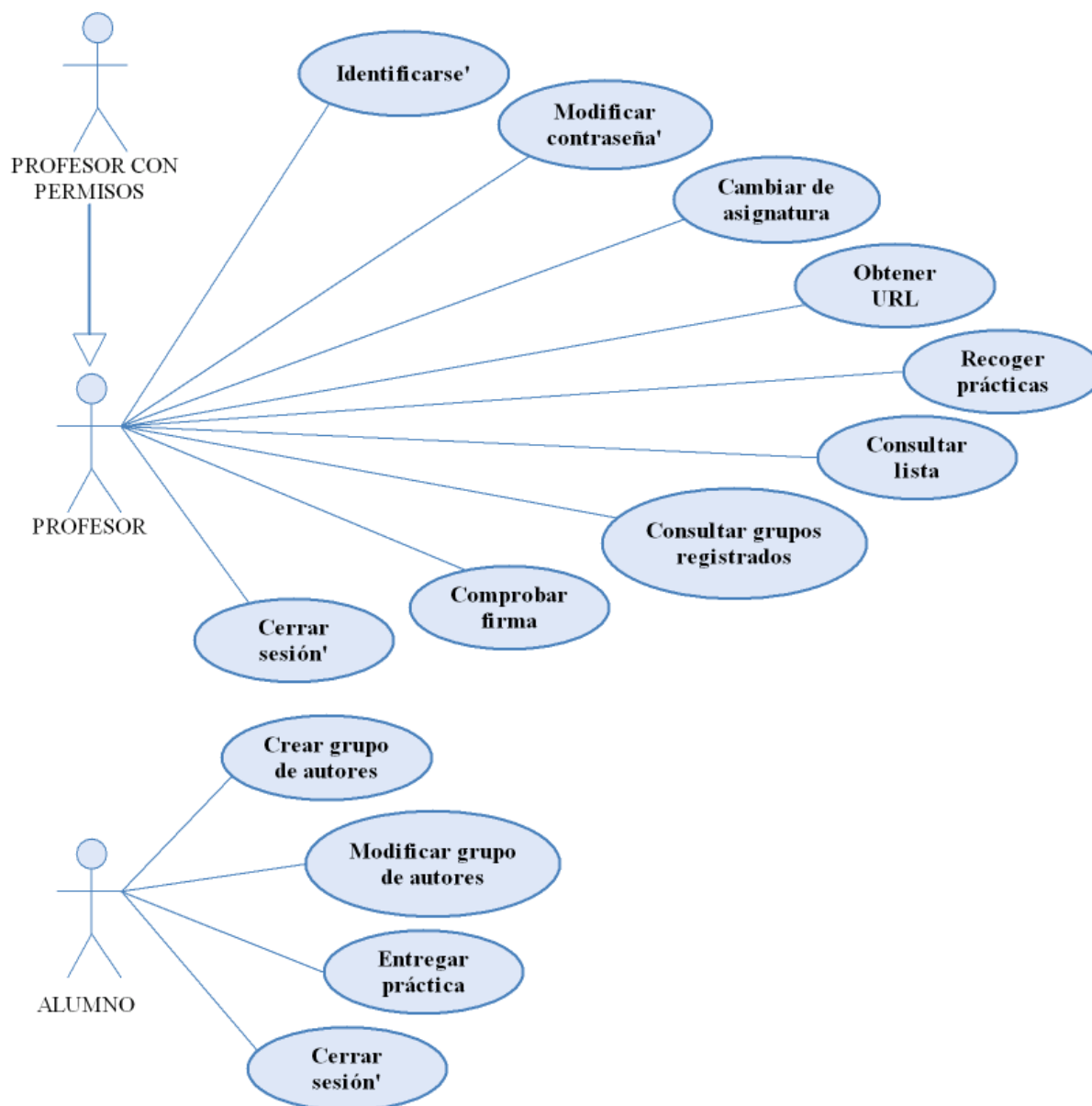
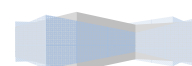


Figura 14: Casos de uso III

#### 4.1.2.- DESCRIPCIÓN DE LOS ACTORES

Los actores son las entidades externas que interactúan con el sistema. En este caso, contamos con los siguientes actores:

- **Administrador:** Persona física que se encarga de instalar el sistema y gestionar los docentes y los coordinadores de cada asignatura.
- **Coordinador:** Persona física encargada de coordinar una asignatura y por lo tanto gestionar todas los profesores de dicha asignatura. Además, puede realizar todas las acciones de un profesor con permisos y de un profesor.



- **Profesor con permisos:** Persona física que además de poder realizar todas las acciones típicas de un profesor, puede gestionar prácticas.
- **Profesor:** Persona física encargada de acciones típicas necesarias para gestionar las correcciones de prácticas, como son recoger las prácticas, corregirlas...
- **Alumno:** Tiene la capacidad de crear y modificar un grupo de prácticas, con la finalidad de poder entregar prácticas de una asignatura automáticamente.

### 4.1.3.- ESPECIFICACIÓN TEXTUAL DE LOS CASOS DE USO

A continuación, se adjuntan las tablas con la especificación textual de cada caso de uso, para su mayor comprensión.

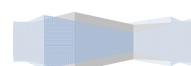
El formato de cada caso de uso será el siguiente:

IDENTIFICADOR:	
Nombre	
Actores	
Objetivo	
Precondiciones	
Postcondiciones	

Tabla 9: Formato de casos de uso

Y se especificarán los siguientes términos:

- **Identificador:** Representa de forma unívoca cada caso de uso. La sintaxis de nombrado será la siguiente: CU-XX, donde XX serán 2 dígitos que numeran ordenadamente los casos de uso, comenzando necesariamente por “CU-01”.
- **Nombre:** Título que se le da a cada caso de uso, de acuerdo a los diagramas del apartado 4.1.1.- *Especificación Gráfica de los Casos de Uso*.
- **Actores:** Usuarios que intervienen directamente en la realización del caso de uso.
- **Objetivo:** Se refiere al objetivo concreto del caso de uso.
- **Precondiciones:** Condiciones que deben darse previamente en el sistema para que el caso de uso pueda efectuarse.
- **Postcondiciones:** Indican el estado del sistema después de la ejecución del caso de uso.



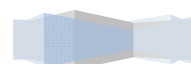


CU-01	
Nombre	Identificarse.
Actores	Administrador, Coordinador, Profesor, Profesor con permisos.
Objetivo	Identificar a un usuario como administrador, coordinador, profesor o profesor con permisos, para que pueda realizar las acciones propias de su condición.
Precondiciones	Ninguna.
Postcondiciones	Administrador, coordinador, profesor o profesor con permisos identificado.

CU-02	
Nombre	Modificar contraseña.
Actores	Administrador, Coordinador, Profesor, Profesor con permisos.
Objetivo	Permite modificar la contraseña de un administrador, coordinador, profesor o profesor con permisos.
Precondiciones	Administrador, coordinador, profesor o profesor con permisos identificado.
Postcondiciones	Contraseña modificada.

CU-03	
Nombre	Activar docente.
Actores	Administrador.
Objetivo	Dar de alta un nuevo docente.
Precondiciones	Administrador identificado.
Postcondiciones	Docente dado de alta.

CU-04	
Nombre	Consultar docente.
Actores	Administrador.
Objetivo	Consultar los datos personales de un docente.
Precondiciones	Administrador identificado.
Postcondiciones	Docente consultado.

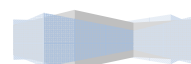


CU-05	
<b>Nombre</b>	Modificar docente.
<b>Actores</b>	Administrador.
<b>Objetivo</b>	Modificar los datos de un docente.
<b>Precondiciones</b>	Docente dado de alta o docente consultado.
<b>Postcondiciones</b>	Docente actualizado.

CU-06	
<b>Nombre</b>	Activar coordinador.
<b>Actores</b>	Administrador.
<b>Objetivo</b>	Dar de alta un nuevo coordinador para una asignatura.
<b>Precondiciones</b>	Administrador identificado.
<b>Postcondiciones</b>	Coordinador dado de alta.

CU-07	
<b>Nombre</b>	Consultar coordinador.
<b>Actores</b>	Administrador.
<b>Objetivo</b>	Consultar los datos personales de un coordinador.
<b>Precondiciones</b>	Administrador identificado.
<b>Postcondiciones</b>	Coordinador consultado.

CU-08	
<b>Nombre</b>	Modificar coordinador.
<b>Actores</b>	Administrador.
<b>Objetivo</b>	Modificar los datos de un coordinador.
<b>Precondiciones</b>	Coordinador dado de alta o coordinador consultado.
<b>Postcondiciones</b>	Coordinador actualizado.

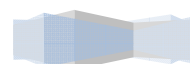


CU-09	
<b>Nombre</b>	Borrar coordinador.
<b>Actores</b>	Administrador.
<b>Objetivo</b>	Eliminar un coordinador.
<b>Precondiciones</b>	Coordinador dado de alta o coordinador consultado.
<b>Postcondiciones</b>	Coordinador borrado.

CU-10	
<b>Nombre</b>	Cerrar sesión.
<b>Actores</b>	Administrador, Coordinador, Profesor, Profesor con permisos y alumno.
<b>Objetivo</b>	Desconectarse de forma segura de la aplicación.
<b>Precondiciones</b>	Administrador, coordinador, profesor, profesor con permisos o alumno identificado.
<b>Postcondiciones</b>	El usuario quedará desconectado de la aplicación y no podrá acceder a sus funcionalidades hasta que no vuelva a identificarse.

CU-11	
<b>Nombre</b>	Activar profesor.
<b>Actores</b>	Coordinador.
<b>Objetivo</b>	Dar de alta a un profesor en una asignatura, además de poderle permitir permisos para gestionar las prácticas.
<b>Precondiciones</b>	Coordinador identificado.
<b>Postcondiciones</b>	Profesor dado de alta.

CU-12	
<b>Nombre</b>	Consultar profesor.
<b>Actores</b>	Coordinador.
<b>Objetivo</b>	Consultar los datos personales de un profesor.
<b>Precondiciones</b>	Coordinador identificado.
<b>Postcondiciones</b>	Profesor consultado.

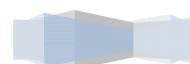


CU-13	
<b>Nombre</b>	Modificar profesor.
<b>Actores</b>	Coordinador.
<b>Objetivo</b>	Modificar los datos de un profesor.
<b>Precondiciones</b>	Profesor dado de alta o profesor consultado.
<b>Postcondiciones</b>	Profesor actualizado.

CU-14	
<b>Nombre</b>	Borrar profesor.
<b>Actores</b>	Coordinador.
<b>Objetivo</b>	Eliminar un profesor.
<b>Precondiciones</b>	Profesor dado de alta o profesor consultado.
<b>Postcondiciones</b>	Profesor borrado.

CU-15	
<b>Nombre</b>	Crear práctica.
<b>Actores</b>	Coordinador, Profesor con permisos.
<b>Objetivo</b>	Crear una nueva práctica para una asignatura y parametrizarla.
<b>Precondiciones</b>	Coordinador identificado o profesor con permisos identificado.
<b>Postcondiciones</b>	Práctica creada.

CU-16	
<b>Nombre</b>	Consultar práctica.
<b>Actores</b>	Coordinador, Profesor con permisos.
<b>Objetivo</b>	Consultar los datos de una práctica.
<b>Precondiciones</b>	Coordinador identificado o profesor con permisos identificado.
<b>Postcondiciones</b>	Práctica consultada.

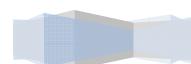


CU-17	
<b>Nombre</b>	Modificar práctica.
<b>Actores</b>	Coordinador, Profesor con permisos.
<b>Objetivo</b>	Modificar los datos de una práctica.
<b>Precondiciones</b>	Práctica creada o práctica consultada.
<b>Postcondiciones</b>	Práctica actualizada.

CU-18	
<b>Nombre</b>	Borrar práctica.
<b>Actores</b>	Coordinador, Profesor con permisos.
<b>Objetivo</b>	Eliminar una práctica.
<b>Precondiciones</b>	Práctica creada o práctica consultada.
<b>Postcondiciones</b>	Práctica borrada.

CU-19	
<b>Nombre</b>	Importar lista.
<b>Actores</b>	Coordinador, Profesor con permisos.
<b>Objetivo</b>	Importa el listado de alumnos de una asignatura en un curso académico.
<b>Precondiciones</b>	Coordinador identificado o profesor con permisos identificado.
<b>Postcondiciones</b>	Lista importada.

CU-20	
<b>Nombre</b>	Consultar lista.
<b>Actores</b>	Coordinador, Profesor y Profesor con permisos.
<b>Objetivo</b>	Consultar los datos de un listado de alumnos.
<b>Precondiciones</b>	Coordinador, profesor o profesor con permisos identificado.
<b>Postcondiciones</b>	Lista consultada.

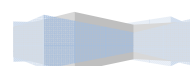


CU-21	
<b>Nombre</b>	Modificar lista.
<b>Actores</b>	Coordinador, Profesor con permisos.
<b>Objetivo</b>	Modificar los datos de un listado de alumnos.
<b>Precondiciones</b>	Lista importada o lista consultada.
<b>Postcondiciones</b>	Lista actualizada.

CU-22	
<b>Nombre</b>	Borrar lista.
<b>Actores</b>	Coordinador, Profesor con permisos.
<b>Objetivo</b>	Eliminar un listado de alumnos.
<b>Precondiciones</b>	Lista importada o lista consultada.
<b>Postcondiciones</b>	Lista borrada.

CU-23	
<b>Nombre</b>	Cambiar de asignatura.
<b>Actores</b>	Coordinador, Profesor, Profesor con permisos.
<b>Objetivo</b>	Permite al coordinador o profesor de una asignatura cambiar a otra de la que también es coordinador o profesor sin tener que cerrar sesión, para poder realizar acciones sobre ellas.
<b>Precondiciones</b>	Coordinador, profesor o profesor con permisos identificado.
<b>Postcondiciones</b>	Asignatura cambiada.

CU-24	
<b>Nombre</b>	Obtener URL
<b>Actores</b>	Coordinador, Profesor, Profesor con permisos.
<b>Objetivo</b>	Obtener la URL de una práctica, con el fin de que los alumnos accediendo a ella puedan realizar las acciones pertinentes por su condición.
<b>Precondiciones</b>	Coordinador identificado, profesor identificado o profesor con permisos identificado.
<b>Postcondiciones</b>	URL obtenida.

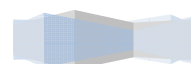


CU-25	
<b>Nombre</b>	Recoger prácticas.
<b>Actores</b>	Coordinador, Profesor, Profesor con permisos.
<b>Objetivo</b>	Descargar la última versión de todas las prácticas entregadas por los alumnos.
<b>Precondiciones</b>	Coordinador identificado, profesor identificado o profesor con permisos identificado.
<b>Postcondiciones</b>	Prácticas descargadas en un directorio.

CU-26	
<b>Nombre</b>	Consultar grupo registrado.
<b>Actores</b>	Coordinador, Profesor, Profesor con permisos.
<b>Objetivo</b>	Consultar los datos de los grupos registrados hasta el momento y las entregas que han realizado,
<b>Precondiciones</b>	Coordinador identificado, profesor identificado o profesor con permisos identificado.
<b>Postcondiciones</b>	Grupo consultado.

CU-27	
<b>Nombre</b>	Comprobar firma.
<b>Actores</b>	Coordinador, Profesor, Profesor con permisos.
<b>Objetivo</b>	Verificar la entrega del alumno.
<b>Precondiciones</b>	Coordinador identificado, profesor identificado o profesor con permisos identificado.
<b>Postcondiciones</b>	Firma comprobada.

CU-28	
<b>Nombre</b>	Crear grupo de autores.
<b>Actores</b>	Alumno.
<b>Objetivo</b>	Crear un nuevo grupo de autores para una práctica.
<b>Precondiciones</b>	Ninguna.
<b>Postcondiciones</b>	Grupo creado.



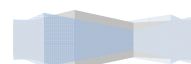
CU-29	
Nombre	Modificar grupo de autores.
Actores	Alumno.
Objetivo	Modificar los datos del grupo de autores.
Precondiciones	Ninguna.
Postcondiciones	Grupo actualizado.

CU-30	
Nombre	Entregar práctica.
Actores	Alumno.
Objetivo	Entregar la práctica de una asignatura electrónicamente.
Precondiciones	Ninguna.
Postcondiciones	Práctica entregada.

## 4.2.- DIAGRAMA DE CLASES

Mediante este tipo de diagrama, se modela la estructura estática del sistema, recogiendo además los atributos y métodos que después se utilizarán para programar la aplicación Web.

El diagrama representa la estructura estática mediante clases y las relaciones existentes entre ellas.





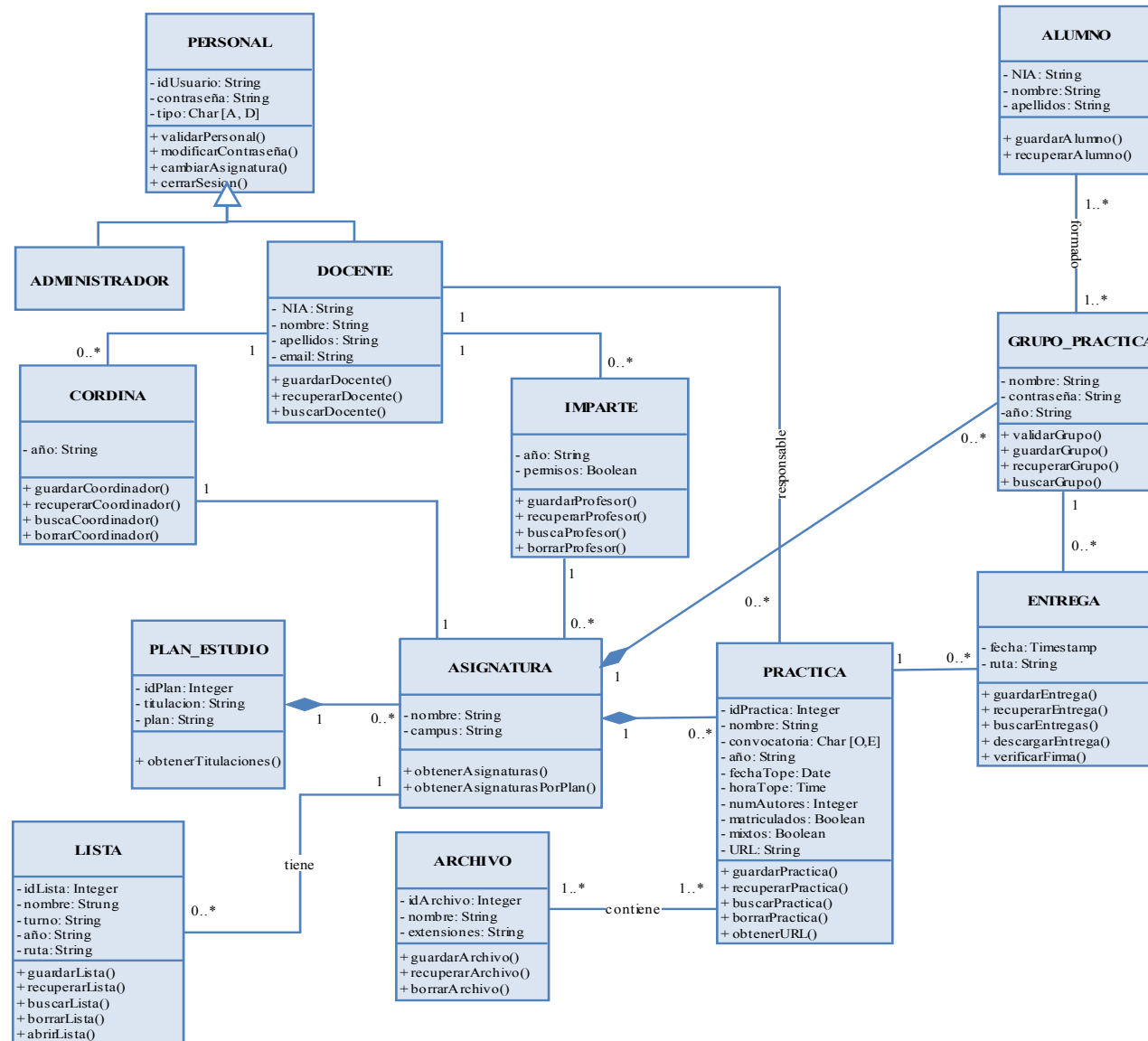
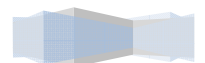


Figura 15: Diagrama de clases



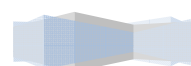
### 4.2.1.- DESCRIPCIÓN DEL DIAGRAMA DE CLASES

---

A continuación, se realizará una breve descripción de cada una de las clases que conforman el diagrama, para después explicar las relaciones existentes entre cada una de ellas.

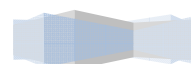
Las clases que forman parte del diagrama son:

- **Personal:** Hace referencia al personal que utiliza la aplicación, en este caso al administrador, los coordinadores y los profesores. Las operaciones de esta clase son:
  - *validarPersonal*: Permite identificar al usuario que accede al sistema otorgándole los permisos pertinentes.
  - *modificarContraseña*: Permite al usuario una vez identificado modificar su contraseña.
  - *cambiarAsignatura*: Permite a los coordinadores y profesores elegir a través de que asignatura quieren acceder al sistema.
  - *cerrarSesion*: Permite al usuario salir de la sesión en el momento que desee.
- **Administrador:** Especialización de la clase personal, que por ello, hereda todos sus atributos y operaciones.
- **Docente:** Especialización de la clase personal que incluye al personal docente de la universidad, es decir, coordinadores y profesores, por ello, hereda todos sus atributos y operaciones. Esta clase además añade nuevos atributos y los siguientes métodos necesarios para realizar acciones sobre los docentes:
  - *guardarDocente*: Permite almacenar los datos de un docente en la BD, además de modificarlos.
  - *recuperarDocente*: Permite recuperar todos los datos de un docente.
  - *buscarDocente*: Permite buscar docentes y recuperar sus datos, dependiendo de unos parámetros de búsqueda.
- **Plan\_Estudio:** Clase que contiene las titulaciones y planes de estudio que forman la universidad. Contiene el método “*obtenerTitulaciones*”, que como su nombre indica, obtiene todas las titulaciones existentes.
- **Asignatura:** Clase que contiene todas las asignaturas impartidas en la universidad, organizadas según el campus que las imparte. Contiene los siguientes métodos:
  - *obtenerAsignaturas*: Obtiene todas las asignaturas existentes.



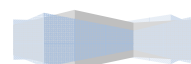


- *obtenerAsignaturasPorPlan*: Obtiene todas las asignaturas correspondientes a un plan de estudios concreto.
- **Coordina**: Clase intermedia que representa la asociación entre un coordinador y la asignatura que coordina. Solo almacena el año académico en el que el coordinador coordina la asignatura. Es considerada clase intermedia, porque pueden existir tuplas repetidas, ya que un coordinador puede coordinar la misma asignatura en diferentes años. Contiene los siguientes métodos:
  - *guardarCoordinador*: Permite almacenar los datos de un docente y la asignatura que coordina en la BD, además de modificarlos.
  - *recuperarCoordinador*: Permite recuperar todos los datos de un docente y la asignatura que coordina.
  - *buscarCoordinador*: Permite buscar docentes y las asignaturas que coordina recuperando sus datos, dependiendo de unos parámetros de búsqueda.
  - *borrarCoordinador*: Elimina la relación existen entre un docente y la asignatura que coordina.
- **Imparte**: Clase intermedia entre un profesor y las asignaturas que imparte, en la cual solo se almacena el año en que se imparte dicha asignatura y un booleano que indica si el profesor tiene o no permisos para gestionar las prácticas. Es considerada clase intermedia, porque pueden existir tuplas repetidas, ya que un profesor puede impartir la misma asignatura en diferentes años. Contiene los siguientes métodos:
  - *guardarProfesor*: Permite almacenar los datos de un docente y la asignatura que imparte en la BD, además de modificarlos.
  - *recuperarprofesor*: Permite recuperar todos los datos de un docente y la asignatura que imparte.
  - *buscarProfesor*: Permite buscar docentes y las asignaturas que imparte recuperando sus datos, dependiendo de unos parámetros de búsqueda.
  - *borrarProfesor*: Elimina la relación existen entre un docente y la asignatura que imparte.
- **Alumno**: Clase que hace referencia a los alumnos que se han registrado alguna vez en un grupo de prácticas. Contiene datos básicos para identificar a los alumnos junto con las siguientes operaciones:
  - *guardarAlumno*: Permite dar de alta nuevos alumnos en la base de datos y/o modificar sus datos.
  - *recuperarAlumno*: Permite recuperar los datos del alumno.





- **Grupo\_Practicas:** Clase que hace referencia a los grupos de prácticas asociados a cada asignatura en un curso académico concreto. Únicamente contiene el nombre del grupo y su contraseña para poder acceder a las acciones típicas de un grupo y el año perteneciente al curso académico, junto con las siguientes operaciones:
  - *validarGrupo*: Permite identificar a un grupo dentro del sistema con el fin de permitirle realizar modificaciones de sus datos y entregar prácticas.
  - *guardarGrupo*: Permite dar de alta un nuevo grupo en la base de datos y modificar sus datos.
  - *recuperarGrupo*: Permite recuperar los datos del grupo.
  - *buscarGrupo*: Permite buscar un grupo.
- **Practica:** Clase que contiene los datos de todas las prácticas dadas de alta en la aplicación junto con operaciones básicas que permite su gestión:
  - *guardarPractica*: Permite almacenar los datos de una práctica en la BD, además de modificarlos.
  - *recuperarPractica*: Permite recuperar todos los datos de una práctica.
  - *buscarPractica*: Permite buscar prácticas y recuperar sus datos, dependiendo de unos parámetros de búsqueda.
  - *borrarPractica*: Elimina una práctica de la BD.
  - *obtenerURL*: Obtiene la URL que los alumnos utilizarán para acceder a una práctica.
- **Entrega:** Clase intermedia entre grupo de prácticas y práctica, en la cual se almacenan la fecha en la que un grupo entrega prácticas y la ruta donde se almacena el fichero. La clase ha sido considerada intermedia ya que admite tuplas repetidas, porque un grupo puede entregar la misma práctica las veces que considere necesarias dentro del plazo estipulado. Contiene los siguientes métodos:
  - *guardarEntrega*: Permite almacenar los datos de una entrega en la BD, además de modificarlos.
  - *recuperarEntrega*: Permite recuperar todos los datos de una entrega.
  - *buscarEntregas*: Permite buscar todas las entregas realizadas por un grupo de prácticas y recuperar sus datos.
  - *descargarEntrega*: Permite al usuario descargar una entrega concreta en su ordenador.



- *verificarFirma*: Permite verificar si la firma electrónica de una entrega concreta se corresponde con la clave pública del servidor.
- **Archivo**: Clase que hace referencia al nombre y extensión de cada uno de los ficheros que debe ser entregado en una práctica.
  - *guardarArchivo*: Permite almacenar los datos de una fichero en la BD, además de modificarlos.
  - *recuperarArchivo*: Permite recuperar todos los datos de fichero.
  - *borrarArchivo*: Elimina un archivo perteneciente a una práctica de la BD.
- **Lista**: Clase que hace referencia a la ruta en la que se encuentran almacenados los listados de alumnos para cada asignatura. Contiene las siguientes funciones:
  - *guardarLista*: Permite almacenar los datos de un listado en la BD, además de modificarlos.
  - *recuperarLista*: Permite recuperar los datos de un listado.
  - *buscarLista*: Permite buscar listados de alumnos y recuperar sus datos.
  - *borrarLista*: Elimina listado asociado a una práctica de la BD.
  - *abrirLista*: Visualiza un listado de alumnos por pantalla.

Las relaciones entre las clases coordinador y asignatura, profesor y asignatura, grupo\_practicas y práctica, quedan especificadas mediante las clases intermedias explicadas anteriormente.

Por otro lado, la relación entre plan\_estudio y asignatura; asignatura y práctica; y asignatura y grupo\_practica es una relación de composición, ya que existe una dependencia total entre ambas partes y por tanto las operaciones de borrado y modificación se propagaran del todo a las partes.

El resto de relaciones son las siguientes:

- **responsable**: Un docente, puede ser o no responsable de una práctica, y una práctica tiene como responsable a un único docente.
- **contiene**: Una práctica puede contener o no varios ficheros entregables, y esos ficheros pertenecen a una única prácticas.
- **formado**: Un grupo de prácticas puede estar formado por al menos un alumno, y un alumno puede pertenecer a uno o varios grupos de prácticas.
- **genera**: Un grupo de práctica genera o no, un único registro, y ese registro solo pertenece a dicho grupo de prácticas.



- **tiene:** Una asignatura puede o no tener listas de alumnos vinculadas, y una lista pertenece a una única asignatura.

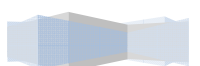
### 4.3.- MODELO RELACIONAL

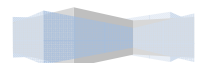
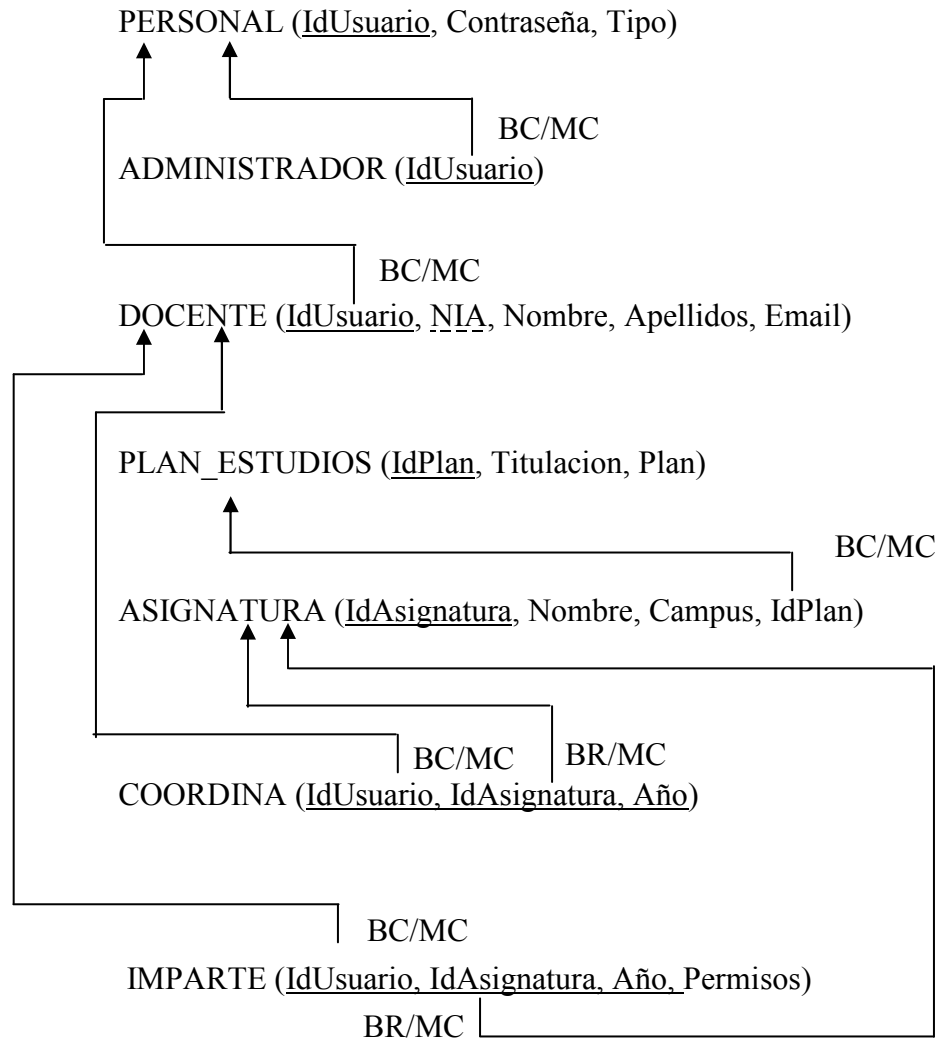
El modelo relacional organiza y representa los datos que se almacenarán en la aplicación, en forma de tablas o relaciones, conformando así la base de datos.

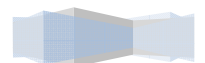
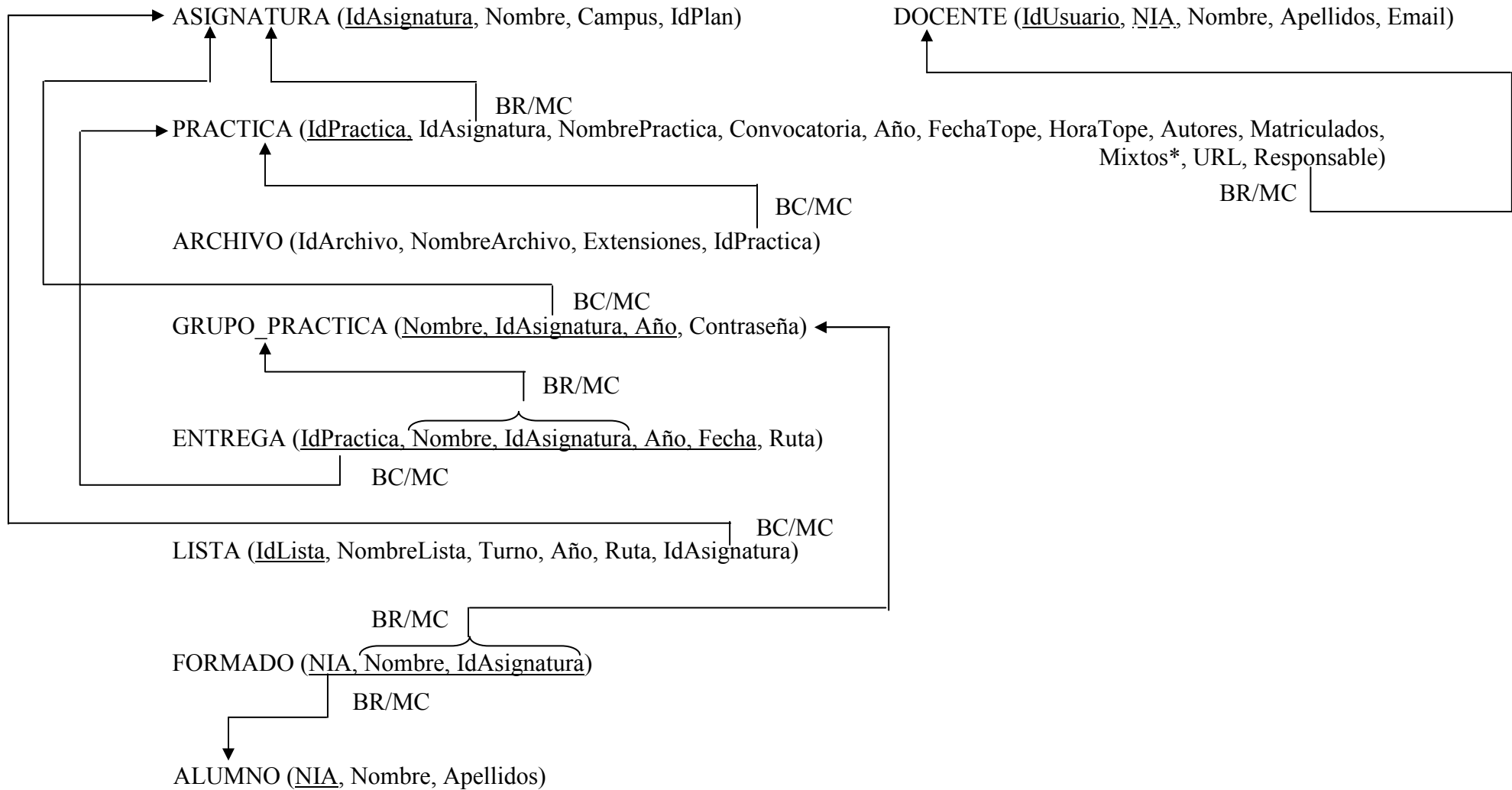
Una relación representa una tabla, donde cada fila representa una colección de valores que describen una entidad del mundo real. Cada fila se denomina tupla o registro y cada columna, campo.

En este modelo, se han subrayado las claves primarias y se han subrayado en discontinuo las claves alternativas.

En cuanto a las modificaciones, siempre se propagan entre las relaciones, mientras que los borrados, únicamente se han restringido en aquellas tablas donde sea necesario almacenar un histórico de los datos.









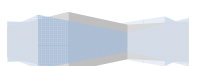
### 4.3.1.- SUPUESTOS SEMÁNTICOS NO RECOGIDOS

---

### 4.3.2.- CHECKS

---

- El contenido del campo “Tipo” de la tabla personal, solo puede contener los siguientes valores:
  - A (Administrador)
  - D (Docente)
- El contenido del campo “Convocatoria” de la tabla práctica, solo puede contener los siguientes valores:
  - O (Ordinaria)
  - E (Extraordinaria)
- El valor del campo “IdUsuario” debe comenzar por una letra.
- El valor del campo “NIA” debe comenzar por 100 y su longitud no excederá de 9 caracteres numéricos.
- El valor del campo “Año” debe mantener el siguiente formato “aa-aa”, el cual indicará el curso académico.
- El valor de los campos “IdPlan”, “IdAsignatura”, “IdPractica”, “IdArchivo”, “IdLista” e “IdGrupo” debe ser un valor mayor que 0.
- El valor del campo “Autores” debe ser un valor mayor o igual que 1.



## 5.- IMPLEMENTACIÓN

En este apartado, se pretende reflejar el funcionamiento de la aplicación Web haciendo hincapié en el modo en el que se ha implementado, comentando por tanto, todos los elementos por los que se encuentra formada.

La documentación de cada uno de los métodos y clases del sistema, se ha generado a través de la herramienta JavaDoc, y por tanto no se incluye en este apartado.

### 5.1.- ARQUITECTURA EN 4 CAPAS

Como ya se comento anteriormente, la aplicación utiliza el patrón MVC a través de su inclusión mediante el marco de trabajo Struts. Además de ello, se utiliza para facilitar el manejo de la BD el marco de trabajo de Hibernate. Todo ello, da lugar a una aplicación constituida por 4 capas, como se muestra en la siguiente figura.

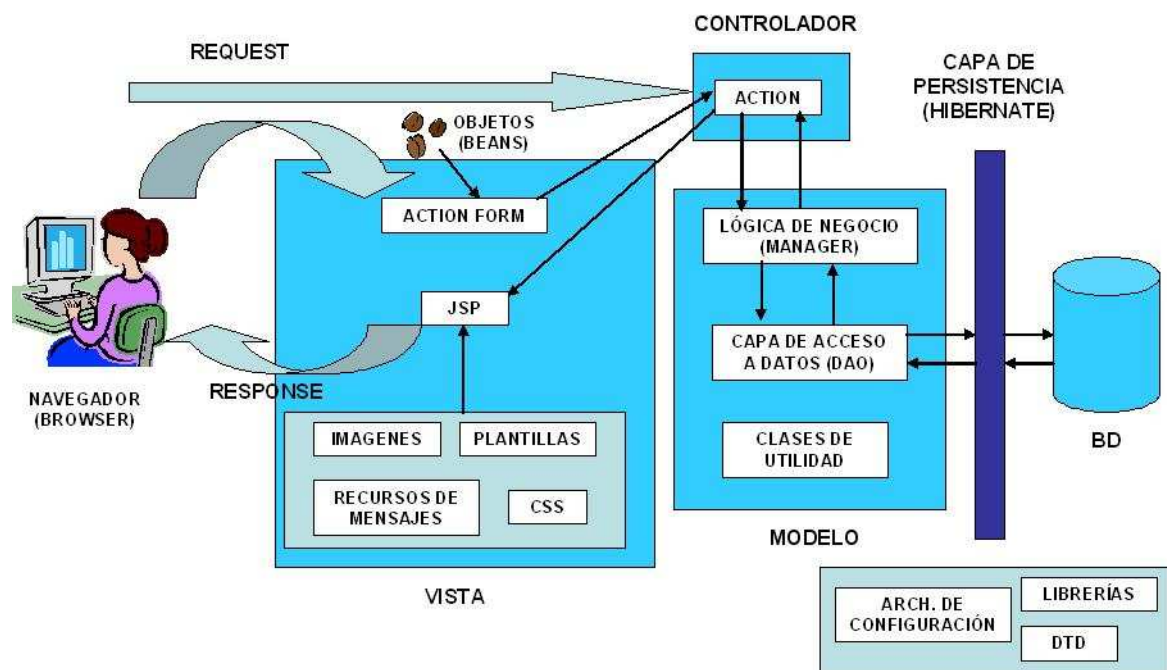
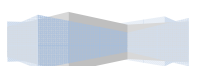


Figura 16: Arquitectura del sistema

Las capas de las que consta el sistema son las siguientes:

- **Vista:** Capa encargada de mostrar el contenido al cliente. En ella, encontramos elementos tales como las páginas Web, las hojas de estilos, imágenes...
- **Controlador:** Capa encargada de recibir las peticiones por parte de los clientes y reenviar dichas peticiones a los elementos de negocio correspondientes. Además se encarga de recibir las respuestas y seleccionar la vista



correspondiente que debe mostrarse al cliente. Dentro de esta capa únicamente encontraremos las clases “Action” que actúan como servlets de la aplicación.

- **Modelo:** Capa que contiene todos los elementos de lógica de negocio de la aplicación. Por ello, dentro de ella podemos encontrar las clases de datos (beans), junto con las clases que contienen los métodos de negocio que permiten acceder a la BD y obtener resultados. En esta aplicación, la parte del modelo hace uso de dos patrones:
  - *Bussines Delegate (Manager)*: Se encarga de crear y gestionar los objetos de lógica de negocio además de proporcionar un interfaz con los métodos de negocio que se pueden utilizar.
  - *Data Access Object (DAO)*: Se utiliza para desacoplar la lógica de negocio del acceso a datos, además de permitir que la implementación de persistencia sea fácilmente reemplazada por otra sin afectar negativamente a los objetos de negocio.
- **Persistencia:** Capa que hace uso del marco de trabajo Hibernate, por lo tanto, dentro de ella, se encuentran los ficheros que mapean los objetos Java a objetos relacionales de la BD, junto con el fichero de configuración del marco de trabajo. Gracias a esta capa, se permite una mayor flexibilidad para trabajar con objetos Java y en la creación de consultas en lenguaje SQL.

A continuación, se explicará con más detalle cada uno de los elementos de cada capa, y por supuesto, el modo en que se comunican unas con otras.

## 5.2.- COMUNICACIÓN Y ELEMENTOS DE LAS CAPAS

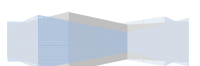
Para explicar la comunicación entre las diferentes capas y como fluye la información dentro de ellas, además de mostrar los aspectos generales, nos centraremos en un ejemplo concreto para facilitar el entendimiento:

*“El administrador de la aplicación tras identificarse en el sistema, desea activar un docente”.*

En primer lugar, cualquier usuario del sistema tendrá opción de acceder al sistema e interactuar con él por medio de los menús de usuario y los diferentes botones que contienen los formularios. En el caso concreto que se ha propuesto, el administrador deberá pulsar en la opción del menú “*Activar docente*”.

Los elementos del menú contienen enlaces a diferentes páginas o servlets que serán cargados siguiendo lo convenido en el fichero de configuración de Struts.

El menú de nuestro ejemplo, es decir, el menú del administrador, muestra el siguiente aspecto:



```
<div class="posicionaMenu fs12px">
  <ul class="menunavegacion" id="menuAdministrador">

  ...

  <!-- Docentes -->
  <li class="menunavegacionli azul">
    <a class="textomenu"> Personal docente</a>
    <ul class="menunavegacion2 azul" id="docentes">
      <li class="menunavegacion2li azul"><a class="textomenuhijo"
        href="/Entregador/recuperarDocente.do"

        onfocus="window.status='Alta';"
        onmouseover="window.status='Alta';return true"
        onmouseout="window.status='';return true" id="altaDocente">
          Activar</a></li>
      <li class="menunavegacion2li azul"><a class="textomenuhijo"
        href="/Entregador/buscarDocente.do"
        onfocus="window.status='Buscar';"
        onmouseover="window.status='Buscar';return true"
        onmouseout="window.status='';return true"
        id="buscarDocente"> Bucar/Modificar/Borrar</a></li>
    </ul>
  </li>

  ...

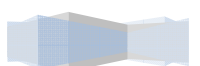
```

Figura 17: menuAdministrador.jsp

El menú, es una JSP, por lo tanto en ella podemos ver el uso de diferentes estilos para mostrar los elementos. La referencia que sigue para activar un docente es “*recuperarDocente.do*”. La aplicación buscará esta etiqueta en el fichero de configuración de Struts para saber que Action o JSP debe cargar.

El fichero de configuración de Struts se encuentra configurado en el fichero “*web.xml*” y dentro de él se encuentran todas las redirecciones necesarias para enviar información dentro de la aplicación y al cliente, además de indicarse las rutas de los paquetes de recursos de mensajes y la configuración de plugins, en nuestro caso el validador.

En nuestro ejemplo, la etiqueta “*recuperarDocente*” del fichero de configuración contiene lo siguiente:



```
<action path="/recuperarDocente"  
        type="actions.docente.RecuperarDocenteAction" name="docenteForm"  
        scope="request" validate="false">  
    <forward name="mostrarDocente"  
            path="/paginas/docente/guardarDocente.jsp" redirect="false"/>  
    <forward name="error" path="/paginas/comunes/error.jsp"  
            redirect="false"/>  
</action>
```

Figura 18: struts-config.xml

Esto indica que:

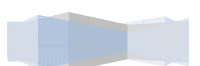
- La petición del cliente será redirigida al Action *RecuperarDocenteAction*, el cual utilizará para recoger y devolver resultados el ActionForm *docenteForm*.
- Tras realizar las acciones pertinentes indicadas en el Action, se mostrará al cliente la página *guardarDocente* o la de error, según se indique en el Action.

### 5.2.1.- ACTION (SERVLET)

Las clases action, son lo que comúnmente se conocen como servlets.

Estas clases recogen las peticiones, realizan todas las acciones necesarias sobre ellas y redirigen los resultados a las páginas o servlets correspondiente.

En nuestro ejemplo, como se comento anteriormente, se hace uso de action *RecuperarDocenteAction*, en el cual podemos encontrar los siguientes elementos:



```
package actions.docente;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import manager.personal.PersonalManager;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import actionForms.DocenteForm;
import beans.Docente;

public class RecuperarDocenteAction extends Action{

public ActionForward execute(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)throws
    Exception {

    DocenteForm formBean = (DocenteForm) form;
    PersonalManager manager = new PersonalManager();

    if (request.getParameter("id") != null &&
        !request.getParameter("id").equals("")){

        Docente docente = manager.obtenerDocente(request.getParameter("id"));
        formBean.setDocente(docente);
        formBean.setContraseña(docente.getContraseña());
        formBean.setInsertar(false);
    }
    else
        formBean.setInsertar(true);

    return mapping.findForward("mostrarDocente");
}

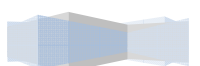
}
```

**Figura 19: RecuperarDocenteAction.java**

El Action o servlet es una clase pública que en este caso extiende de la clase Action de Struts y por lo tanto hereda sus métodos y atributos.

Dentro de esta clase sobrescribimos el método “*execute*” indicándole lo siguiente:

- Instanciamos el ActionForm correspondiente que se incluía en el fichero de configuración, es decir, *DocenteForm*.
- Instanciamos *PersonalManager*, que es la clase que contiene los métodos negocio para los docentes, y que pertenece al patrón indicado anteriormente Business Delegate.
- Preguntamos si hemos recibido en la petición el identificador de usuario en cuyo caso deberemos obtener los datos del docente a través del manager y enviarlos a través del ActionForm.



- Si no hemos recibido el identificador de usuario, que será nuestro caso, modificaremos el atributo “insertar” del ActionForm poniendolo a true.
- Finalmente en cualquiera de los casos anteriores, redireccionaremos a la etiqueta “*mostrarDocente*” que en el archivo de configuración nos enviaría a la página guardarDocente.

### 5.2.2.- LA JSP

Después de procesarse una petición, generalmente el action suele redirigirla a una página Web, aunque a veces también suele pasarse a otro action y finalmente a una página Web.

En nuestro ejemplo, el action nos redirigirá a la página *guardarDocenteBody*, la cual muestra el siguiente aspecto:



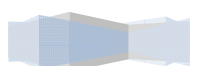
Figura 20: Jsp guardar docente

Todas las páginas de la aplicación se han construido haciendo uso del patrón “*Tiles*”, que forma parte de Struts.

Este patrón evita la repetición de contenido por medio del uso de plantillas y definiciones, de esta manera los elementos comunes como pueden ser la cabecera, subcabecera y el menú que deben aparecer siempre, se encuentran en distintas páginas y se cargan por medio de una plantilla y una definición consiguiendo de esta manera que dichos elementos sean estáticos y haciendo que el único contenido dinámico sean los formularios.

A continuación se muestra un ejemplo del uso de “*Tiles*”:

- Primero debemos crear una definición para cada tipo de usuario. Las definiciones permiten especificar estáticamente el contenido de una plantilla. Dentro de la definición, indicamos el lugar donde se encuentra el contenido estático a cargar, en nuestro caso, podemos observar como se carga el título principal de las páginas, la cabecera, la subcabecera, el menú del administrador y los mensajes.

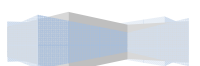


```
<%@ taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles"%>

<tiles:definition
  id="administrador.principal"
  page="/plantillas/plantillaAdministrador.jsp"
  scope="request">
  <tiles:put name="titulo.principal" value="Entregador de prácticas"/>
  <tiles:put name="cabecera" value="/comun/cabecera.jsp" />
  <tiles:put name="subcabeceraAdministrador"
value="/comun/subcabeceraAdministrador.jsp"/>
  <tiles:put name="menuAdministrador" value="/comun/menuAdministrador.jsp" />
  <tiles:put name="mensajes" value="../comun/mensajes.jsp" />
</tiles:definition>
```

**Figura 21: administrador-defs.jsp**

- Después debemos crear una plantilla para cada definición, en la que indiquemos la cabecera de la JSP junto con el cuerpo.
  - *Cabecera*: Aquí indicamos las hojas de estilo que vamos a usar, concretamente en nuestro caso, le indicamos que según sea la resolución del usuario cargue una hoja de estilos u otra. Además se indica el tipo de contenido, el nombre de todas las páginas y que cargue un calendario definido a través de funciones JavaScript .
  - *Cuerpo*: En él, se define por medio de etiquetas que indican el contenido que existirá, pero sin indicar donde se encuentra.





```
<%@ taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <script type="text/javascript" language="javascript">
      if ( screen.width == 1024){
        document.write ('<link rel="stylesheet"
          href="/Entregador/css/estilos1024.css">');
      }
      else {
        document.write ('<link rel="stylesheet"
          href="/Entregador/css/estilos1280.css">');
      }
    </script>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title><tiles:getAsString name="titulo.principal" /></title>
    <link rel="stylesheet" href="/Entregador/css/calendario.css"
      type="text/css"/>
    <script type="text/javascript" src="/Entregador/js/calendar.js"></script>
    <script type="text/javascript" src="/Entregador/js/calendar-
      setup.js"></script>
    <script type="text/javascript" src="/Entregador/js/calendar-
      es.js"></script>
  </head>
  <body>
    <tiles:insert attribute="cabecera" />
    <tiles:insert attribute="subcabeceraAdministrador"/>
    <div style="padding-top:1.5em;">
      <tiles:insert attribute="menuAdministrador" />
      <div class="posicionaContenido1 fs12px">
        <tiles:insert attribute="mensajes" ignore="true"/>
        <tiles:insert attribute="contenido"/>
      </div>
    </div>
  </body>
</html>
```

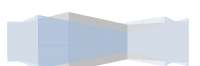
Figura 22: plantillaAdministrador.jsp

- Finalmente se declaran todas y cada una de las páginas JSP indicadas en la definición, además del contenido de cada página que es el único elemento dinámico.  
Para facilitar este trabajo, se utilizan dos páginas, una que contiene la inclusión del resto de elementos y que indica el contenido a cargar, y por otro lado otra página con el contenido dinámico, de modo que si fuera necesario realizar algún cambio fuera más sencillo.

```
<%@ taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles"%>
<%@include file="../../comun/administrador-defs.jsp" %>

<tiles:insert beanName="administrador.principal" beanScope="request">
<tiles:put name="contenido" value="/paginas/docente/guardarDocenteBody.jsp"/>
</tiles:insert>
```

Figura 23: guardarDocente.jsp



Ya en la página JSP del contenido, podemos ver como se declara el elemento `ActionForm` a utilizar, indicando también el `Action` donde debe redirigirse la petición y por supuesto indicando que el formulario se validará a través del plugin `Validator` de Struts. Tras esto, se declaran cada uno de los elementos haciendo uso de `TagLibraries`, y los tags de Struts.

```
<div class="contenido">

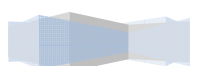
    <html:javascript formName="docenteForm"/>
    <html:form action="/guardarDocente" onsubmit="return
        validateDocenteForm(this);">
    ...
        <fieldset>
            <legend class="negrita">Datos del docente</legend>
            <div class="fondoFormulario">
                <table class="tabla"><tbody>
                    <tr>
                        <td class="textolabelizq"><label for="id">Identificador de
                            usuario*: </label></td>
                        <td class="negrita">
                            <c:choose>
                                <c:when test="${!empty docenteForm.docente.idUsuario}">
                                    <c:out value="${docenteForm.docente.idUsuario}"/>
                                </c:when>
                                <c:otherwise>
                                    <html:text styleId="id" property="docente.idUsuario"
                                        maxLength="5"/>
                                </c:otherwise>
                            </c:choose>
                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
    ...
</div>
```

Figura 24: guardarDocenteBody.jsp

Gracias al validador que se encuentra configurado dentro de *WEB-INF/validation.xml*, si el usuario no introduce algún campo obligatorio o el valor del campo es incorrecto, por medio de JavaScript se le avisará del error, no permitiendo enviar el formulario al servidor hasta que no sea correcto.

```
<field property="docente.contraseña" depends="required,minlength">
    <arg key="campo.pwd"/>
    <arg name="minlength" key="${var:minlength}" resource="false"/>
    <var>
        <var-name>minlength</var-name>
        <var-value>7</var-value>
    </var>
</field>
```

Figura 25: validation.xml



### 5.2.3.- EL MANAGER

Dentro de las clases action, se suelen instanciar las clases de negocio necesarias para realizar acciones sobre los datos.

En nuestro ejemplo, y suponiendo que estamos trabajando con la inserción de un nuevo docente, el action instanciará la clase de negocio *PersonalManager* y llamará al método “*insertarOActualizarDocente*”.

```
public String insertarOActualizarDocente(Docente docente) throws Exception {  
    Session session = dao.obtenerSesion();  
    dao.insertarOActualizarObjeto(docente, session);  
    return docente.getIdUsuario();  
}
```

Figura 26: PersonalManager.java

Este método de negocio, abrirá una nueva sesión en la BD y llamará al método “*insertarOActualizarObjeto*” de la capa de acceso a datos, devolviendo finalmente el identificador del usuario insertado.

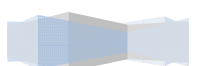
### 5.2.4.- LA CAPA DE ACCESO A DATOS (DAO)

En la clase Dao, tenemos todos los métodos necesarios para interactuar con la BD y hacer uso del marco de trabajo Hibernate.

En nuestro ejemplo concreto, tras abrir una sesión en la BD, se insertará el objeto correspondiente por medio de una transacción.

```
public void insertarOActualizarObjeto(Object o, Session session) throws  
Exception {  
    Transaction tx = null;  
    tx = session.beginTransaction();  
    session.saveOrUpdate(o);  
    tx.commit();  
    session.flush();  
    session.close();  
}
```

Figura 27: Dao.java



### 5.2.5.- HIBERNATE

La interacción con la BD se realiza por medio del marco de trabajo Hibernate. Este marco contiene la configuración de la BD, junto con el mapeo de objetos Java a objetos relacionales. Además permite hacer uso de un lenguaje específico para crear consultas sobre la BD.

Los ficheros de mapeo, son ficheros xml, donde se especifica la correspondencia entre una clase Java y la tabla de la BD a la que hace referencia, los atributos de cada objeto y su correspondiente columna de la BD y por supuesto las relaciones que mantiene dicho objetos con otros.

En el ejemplo, la clase docente hereda de la clase personal, por lo tanto el fichero de mapeo de dicha clase se encuentra contenido en el fichero de la clase personal.

Como se puede observar a continuación, la clase personal tiene como identificador el idUsuario, además de dos atributos: tipo y contraseña. Por otro lado, contiene dos clases hijas administrador y docente cada uno con sus identificadores y atributos correspondiente.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping package="beans" default-lazy="false">
  <class name="Personal" table="personal">
    <id name="idUsuario" column="id_usuario">
      <generator class="assigned"/>
    </id>
    <property name="contraseña" column="contraseña"/>
    <property name="tipo" column="tipo"/>
  </class>
  <joined-subclass name="Administrador" table="administrador"
    extends="Personal">
    <key column="id_usuario"/>
  </joined-subclass>
  <joined-subclass name="Docente" table="docente" extends="Personal">
    <key column="id_usuario"/>
    <property name="nia" column="nia"/>
    <property name="nombre" column="nombre"/>
    <property name="apellidos" column="apellidos"/>
    <property name="email" column="email"/>

    <set name="coordina" table="coordina" inverse="true">
      <key column="id_usuario" not-null="true"/>
      <many-to-many class="Asignatura" column="id_asignatura"/>
    </set>
  </joined-subclass>
</hibernate-mapping>
```

Figura 28: Personal.hbm.xml





### **5.2.6.- EL PAQUETE DE RECURSOS DE MENSAJES**

---

La aplicación cuenta también con un fichero denominado paquete de recursos de mensajes. Este fichero contiene constantes que permiten mostrar al usuario mensajes concretos de la aplicación.

Por ejemplo, cada vez que salta el validador, este recoge el nombre de los campos de dicho paquete. Por otro lado, los mensajes de error y validación dentro de la aplicación también se encuentran definidos en este paquete.



## 6.- VERIFICACIÓN DEL SISTEMA

Para asegurarse del correcto funcionamiento de la aplicación, esta ha sido sometida a diversas pruebas, cuyos resultados han sido positivos.

En primer lugar se han realizado pruebas de caja blanca, que son pruebas software que se realizan sobre las funciones internas de un módulo, de manera que se consigan cubrir todo el código del programa comprobando los posibles errores.

En segundo lugar se han realizado pruebas de caja negra, que son pruebas funcionales que se realizan sobre el exterior de un módulo, en las que se aporta una entrada y se comprueba la salida, sin importar lo que haya ocurrido dentro del módulo

Finalmente se han realizado pruebas de accesibilidad para comprobar si la interfaz cumple con los requisitos de la W3C.

A continuación se muestra el plan de pruebas seguido.

### 6.1.- PRUEBAS DE CAJA BLANCA

Durante la etapa de desarrollo, el programa se ha compilado diversas veces, comprobando los resultados obtenidos tras su ejecución en los dos entornos donde debe funcionar, Windows y Linux, además de en dos navegadores Explorer y Mozilla.

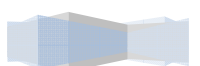
Una vez terminada la aplicación, esta ha sido sometida a nuevas pruebas, entre ella se han realizado pruebas introduciendo siempre datos ficticios y comenzando con una BD extensa para ver si esto causaba algún error grave o tiempo alto de respuesta, consiguiendo un resultado satisfactorio.

Otra de las pruebas ha sido mediante debugging comprobar que todos los posibles caminos que existen en el código fueran recorridos y no dejarse así algo sin verificar.

### 6.2.- PRUEBAS DE CAJA NEGRA

Se han llevado a cabo pruebas sobre los datos de entrada, comprobando especialmente campos numéricos, fechas y horas, de manera que se introducían datos incorrectos para ver el funcionamiento de la aplicación en estos casos, y también valores correctos, con el fin de probar todas las posibles opciones. El sistema actúa de manera correcta cuando se introducen datos incorrectos, ya que el usuario es avisado en todo momento de su error, de modo que pueda rectificar su acción.

En cuanto al aspecto externo de la aplicación, siguiendo los requisitos especificados, no se visualiza ningún problema ejecutándola sobre los navegadores Explorer 8 y Mozilla 3, mediante una resolución de pantalla de 1024\*768 o 1280\*1024. En el momento que dicha resolución es modificada, el contenido de la aplicación (formularios) se puede desplazar impidiendo visualizarlos de una forma correcta.



Finalmente la aplicación ha sido probada mediante la conexión simultánea de varios usuarios, intentado así simular su funcionamiento real, obteniendo unos tiempos de respuesta cortos y sin afectar el rendimiento de la BD.

### 6.3.- PRUEBAS DE ACCESIBILIDAD

El Consorcio para la World Wide Web (W3C) fue fundado en octubre de 1994 para conducir a la World Wide Web (WWW) a su máximo potencial desarrollando protocolos de uso común que promocionaran su evolución y aseguraran su interoperabilidad.

La accesibilidad de un sitio Web se refiere a su capacidad para presentar contenidos a personas que cuentan con discapacidades físicas, que les impiden usar la información disponible de una manera tradicional y por ello emplean ayudas técnicas.

Para verificar la accesibilidad de una Web, la Web Accessibility Initiative (WAI) ha creado una serie de pautas de accesibilidad, que según su grado de prioridad darán a la página un mayor o menor nivel de accesibilidad.

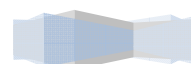
Las prioridades establecidas son las siguientes:

- **Prioridad 1 (A):** Un desarrollador tiene que satisfacer este punto de verificación, ya que sino, uno o más grupos de usuarios encontrarán imposible acceder a la información del documento. Satisfacer este punto de verificación es un requerimiento básico para que algunos grupos puedan usar los documentos Web.
- **Prioridad 2 (AA):** Un desarrollador debe satisfacer este punto de verificación, de otra forma, uno o más grupos encontrarán dificultades en el acceso a la información del documento. Satisfacer este punto de verificación eliminará importantes barreras de acceso a los documentos Web.
- **Prioridad 3 (AAA):** Un desarrollador puede satisfacer este punto de verificación, en caso contrario, uno o más grupos de usuarios encontrarán alguna dificultad para acceder a la información del documento. Satisfacer este punto de verificación mejorará la accesibilidad de los documentos Web.

En la realización de este proyecto, se ha buscado cumplir al menos la prioridad 1 en todas las páginas, y siempre que ha sido posible también cumplir las prioridades 2 y 3.

Para analizar el grado de accesibilidad del proyecto, se han utilizado tres tipos de validadores y analizadores:

- Analizador de contrastes: Para validar el color de las páginas.
- Validador de hojas de estilo en cascada.
- Test de accesibilidad Web.



A continuación se explicarán los test realizados y los resultado obtenidos.

### 6.3.1.- ANALIZADOR DE CONTRASTES

Es muy importante, asegurarse de que los textos y gráficos son comprensibles cuando se ven sin color, ya que algunas personas no pueden diferenciar bien todos los colores ó confunden unos con otros. Además, aunque no es lo común, algunos usuarios pueden no tener pantallas a color.

Cuando los colores de primer plano y de fondo tienen un tono similar, pueden no proporcionar suficiente contraste en las pantallas monocromáticas, así como a las personas con diferentes tipos de deficiencias de percepción de los colores.

Para poder verificar este aspecto, se ha utilizado un analizador de contrastes antes de utilizar un color en cada página.

Dicho programa puede descargarse gratuitamente desde la siguiente dirección:

<http://www.visionaustralia.org.au/info.aspx?page=959>

Se muestra un ejemplo del resultado obtenido al comparar el color de fondo de los formularios Web y el de los textos.

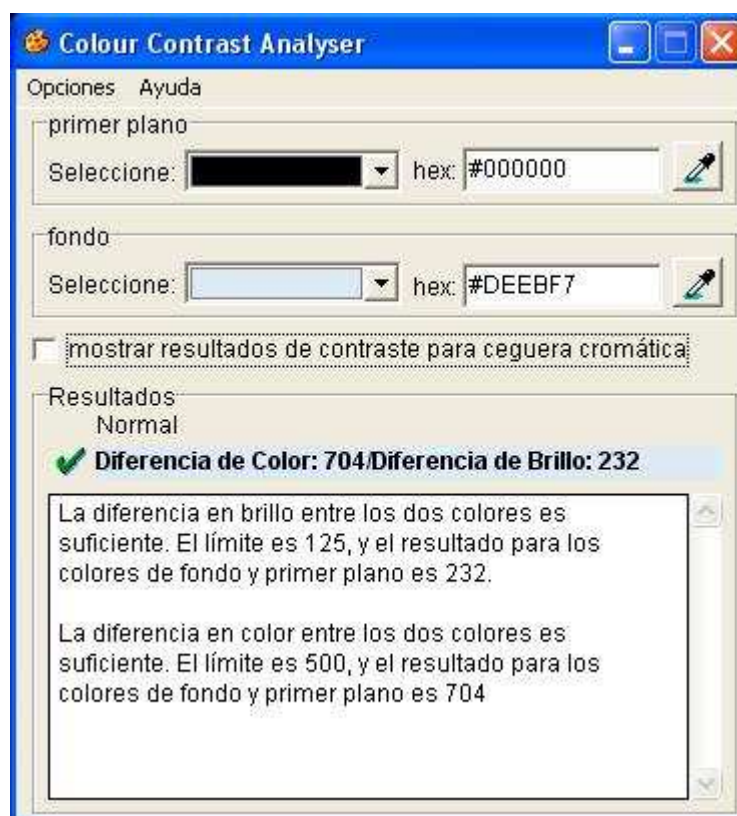
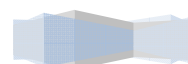


Figura 29: Analizador de contrastes I





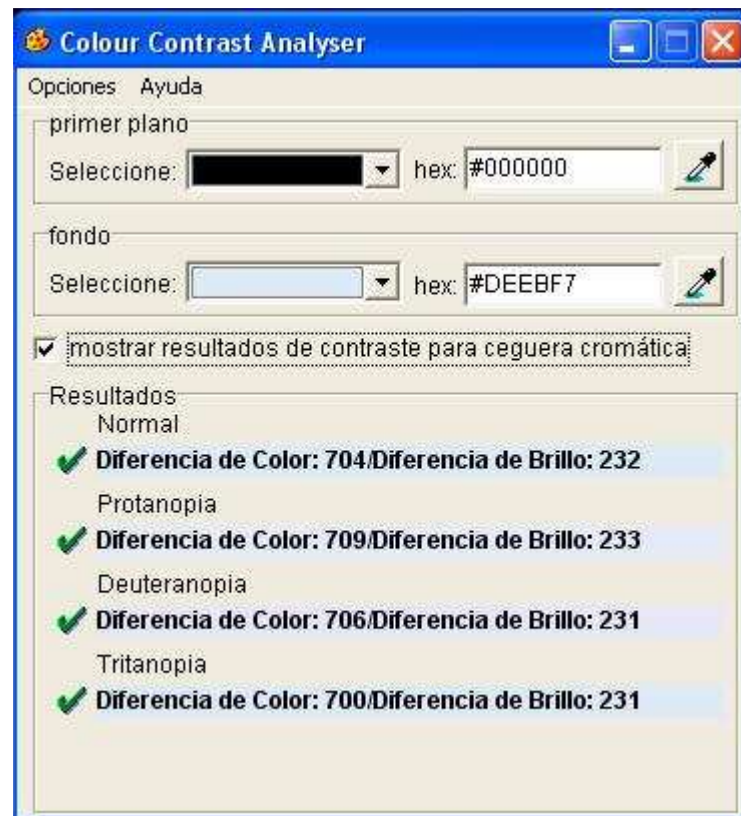


Figura 30: Anaizador de contrastes II

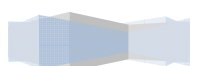
Como puede observarse se utilizan colores que se pueden visualizar correctamente en caso de problemas de visión.

### 6.3.2.- VALIDADOR DE HOJAS DE ESTILO

Para verificar las hojas de estilo del proyecto, es necesario introducir la siguiente dirección en un navegador Web:

*<http://jigsaw.w3.org/css-validator/>*

Tras ello, se cargara en el navegador la siguiente página, la cual contiene el validador de CSS.



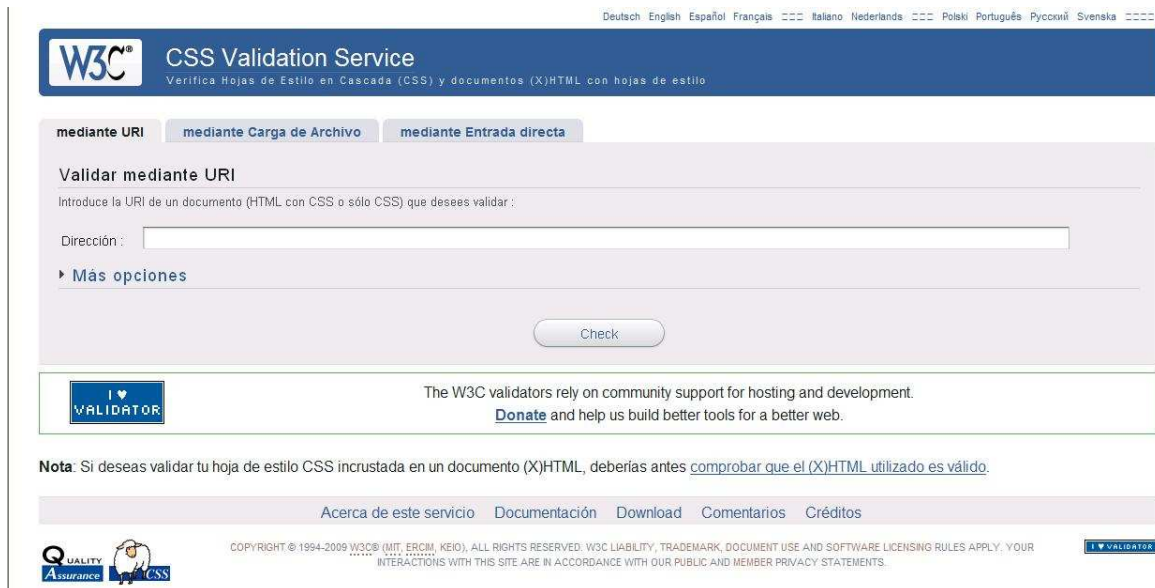


Figura 31: Validador de CSS

Para validar las hojas de estilo del proyecto, se han cargado mediante la opción “Carga de archivos” y se ha indicado en el perfil la versión 2 de CSS. Tras realizar el análisis se obtuvieron resultados satisfactorios, para todas las CSS utilizadas.



Figura 32: Resultado validador de CSS

### 6.3.3.- TEST DE ACCESIBILIDAD WEB

Para realizar el análisis de accesibilidad de todas las páginas del proyecto, se ha utilizado el programa Test de Accesibilidad Web 3.08 (TAW), desarrollado por el ministerio de Industria, Turismo y Comercio; el Gobierno del Principado de Asturias y la fundación CTIC.

Este programa se puede descargar de forma gratuita o utilizarse de manera on-line a través de la siguiente dirección:

*<http://www.tawdis.net/>*

Como ha sido necesario analizar diversas páginas, se ha descargado el programa y se han analizado una a una todas las páginas del proyecto.



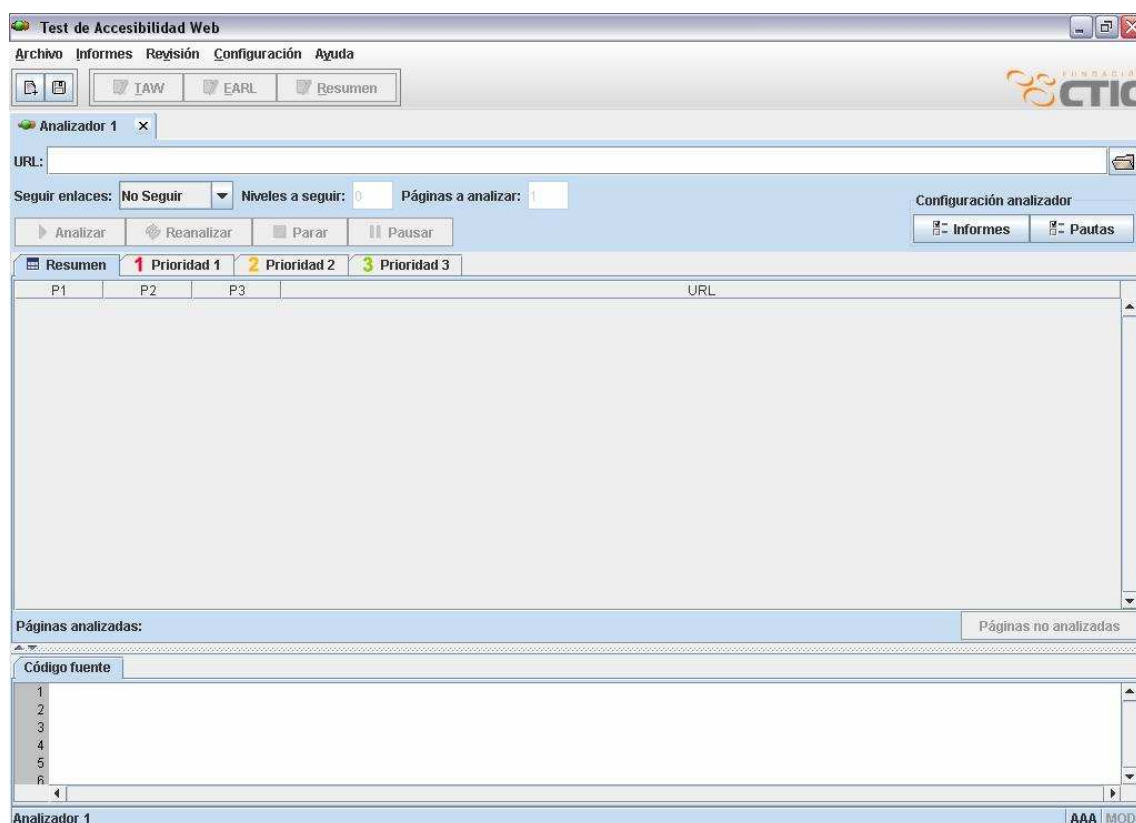


Figura 33: Test de accesibilidad

Para analizar cualquier página, solo se necesita introducir la ruta de dicha página ó la ruta del archivo en formato .html, en “URL” y pulsar sobre el botón “Analizar”, de forma que debajo aparecerán los resultados, y en cada pestaña se verán desglosados estos, por nivel de accesibilidad.

Desde el comienzo se ha intentado realizar páginas accesibles, por lo que tras pasar el test, no se han observado muchos errores.

A continuación se explican los errores encontrados y las soluciones aportadas:

➤ **Prioridad 1 (A):**

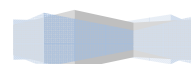
- 1.1. Proporcione un texto equivalente para todo elemento no textual, es decir, que a todas las imágenes debe añadirse la etiqueta “alt” para dar una descripción.

➤ **Prioridad 2 (AA):**

- 11.2. Evite características desaconsejadas por las tecnologías W3C, este error se produce por las etiquetas JavaScript del tipo

```
<script type="text/javascript" language="javascript">
```

Por lo tanto, la solución es la siguiente:



```
<script type="text/javascript">
```

A pesar de ello, el marco Validator de Struts, introduce dinámicamente las etiquetas, por lo tanto, aquí no es posible modificarlo, por ello, las páginas que no superan la AA, es porque hacen uso del validador.

- 12.4. Asocie explícitamente las etiquetas con sus nombres, el problema residía en que cuando se utilizaban campos de selección (combobox) no se añadía el identificador que referenciaba a la etiqueta “label”.

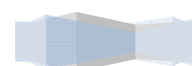
➤ **Prioridad 3 (AAA):**

- 5.5. Proporcione resúmenes a la tablas, es decir, se debe añadir “summary” a los elemento de tipo “table”.
- 5.6. Proporcionar abreviaturas para las etiquetas de encabezamiento, es decir que en los encabezados de las tablas que son demasiado extensos debe añadirse la etiqueta “abbr” con una abreviatura para dicho encabezado.
- 10.4. Hasta que las aplicaciones de usuario manejen correctamente los controles vacíos, incluya caracteres por defecto en los cuadros de edición y áreas de texto. Este error no se ha logrado solventar, y tras leer en diversos sitios, no queda claro si esta o no deprecado, por lo tanto todas las páginas que no superan la AAA es debido a este error.

Tras el análisis de todas las páginas, estos son los resultados obtenidos:

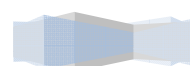
PÁGINA	A	AA	AAA
buscarCalificacionBody	Sí	Sí	Sí
buscarCoordinadorBody	Sí	Sí	No
buscarDocenteBody	Sí	Sí	No
buscarEntregasBody	Sí	Sí	No
buscarGrupoBody	Sí	Sí	Sí
buscarListaBody	Sí	Sí	Sí
buscarPracticaBody	Sí	Sí	No
buscarProfesorBody	Sí	Sí	No
buscarUrlBody	Sí	Sí	Sí
datosGrupoBody	Sí	Sí	Sí
descargarPracticaBody	Sí	Sí	Sí
entregarPracticaBody	Sí	No	No

Tabla 10: Resultados accesibilidad I



PÁGINA	A	AA	AAA
errorBody	Sí	Sí	Sí
guardarAlumnoBody	Sí	No	Sí
guardarArchivoBody	Sí	No	No
guardarCoordinadorBody	Sí	No	No
guardarDocenteBody	Sí	No	No
guardarGrupoBody	Sí	No	Sí
guardarListaBody	Sí	No	Sí
guardarPracticaBody	Sí	Sí	No
guardarProfesorBody	Sí	No	No
inicioAlumnoBody	Sí	Sí	Sí
inicioBody	Sí	No	No
inicioCoordinadorBody	Sí	Sí	Sí
inicioDocenteBody	Sí	Sí	Sí
inicioProfesorBody	Sí	Sí	Sí
inicioProfesorPermisosBody	Sí	Sí	Sí
modificarPwdBody	Sí	No	Sí
validarGrupoBody	Sí	No	No
verificarFirmaBody	Sí	Sí	No
visualizarCalificacionBody	Sí	Sí	Sí
visualizarListaBody	Sí	Sí	Sí

Tabla 11: Resultados accesibilidad II



## 7.- IMPLANTACIÓN

En este apartado se describen los pasos necesarios para la correcta instalación de todos los programas necesarios para el funcionamiento de la aplicación.

Se destaca, que la aplicación es multiplataforma, es decir, que ha sido implementada tanto para su despliegue en entornos Windows como Linux. Además de ello, existen dos modos de instalación:

- **Modo estándar:** Simplemente se despliega la aplicación en el servidor, sin modificar ninguna parte del código y a través de un fichero .war.
- **Modo personalizado:** Se configuran los parámetros necesarios y se genera el .war que después deberá ser desplegado en un servidor.

A continuación se muestran todos los requisitos de implantación del sistema para un entorno Linux. El plan de implantación para entornos Windows, se encuentra en el *Anexo II.- Implantación en Entornos Windows*.

### 7.1.- FORMACIÓN NECESARIA PARA LA IMPLANTACIÓN

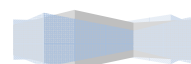
Sería recomendable que el sistema fuera implantado por el diseñador de la misma ya que este, cuenta con todos los conocimientos necesarios para llevarla a cabo sin ninguna complicación.

A pesar de ello, y gracias a los siguientes apartados y explicaciones, cualquier persona con unos conocimientos avanzados en informática, puede ser capaz de realizar la implantación.

Los conocimientos necesarios que se presuponen para realizar la implantación son:

- Sistemas software y tecnologías implicadas: instalación, configuración y administración del servidor de aplicaciones Apache Tomcat, del servidor de base de datos MySQL y de la máquina virtual de Java. Por otro lado, conocimientos generales de Linux.
- Conocimiento del uso de la aplicación y su configuración, que puede obtenerse en las siguientes líneas y leyendo los manuales de usuario que se encuentran en el *Anexo I.- Manuales de Usuario*.

Tras la instalación, se recomienda asignar a un docente como administrador del sistema de manera que además de acceder al sistema con dicho rol, pueda acceder como coordinador de alguna asignatura y probar la aplicación y aprender su manejo. Si el docente no detecta ningún problema, se podrán incorporar el resto de asignaturas de las titulaciones de la universidad y dicho docente puede formar al resto del personal que por supuesto también debería leer los manuales de usuarios del *Anexo I.- Manuales de Usuario*.



Los alumnos por el contrario, deberían leer el manual de usuario descrito para ellos y que se encuentra en el *Anexo I.- Manuales de usuario / 3.- Manual de Usuario de Alumnos*. A pesar de ello, se recomienda que los profesores hagan una demostración del uso de la aplicación por parte de los alumnos para aclarar posibles dudas.

## 7.2.- INFRAESTRUCTURA

En cuanto a la infraestructura física, las instalaciones del cliente destinadas a albergar el servidor deben reunir las condiciones adecuadas para contener equipos que estarán permanentemente en funcionamiento:

- Protección contra incendios.
- Control ambiental (climatización).
- Acceso controlado.
- Cables y dispositivos ordenados y correctamente etiquetados.
- Sistemas de alimentación ininterrumpida.
- Suministro eléctrico protegido.

## 7.3.- ASPECTOS HARDWARE

Para poner en marcha la aplicación, será necesario implantarla en el centro de cálculo de ARCOS ya sea en una máquina virtual del servidor central o incluso en el propio servidor central, lo cual se deja a elección del cliente.

Dicho servidor deberá contar al menos con las siguientes características:

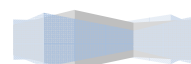
- Procesador Intel Core 2Duo a 2Ghz.
- 2 GB de memoria RAM.
- Disco duro de 200GB.
- Lector de CD.

Además, deberá contarse con un servidor de respaldo, en el que puedan realizarse backups periódicos tanto del sistema como de los datos contenidos en la BD. La asiduidad del backup se dejará a elección del cliente.

## 7.4.- ASPECTOS SOFTWARE

La aplicación ha sido probada en equipos que contaban con la distribución de Linux Ubuntu 9.04, por lo tanto, se recomienda su instalación en dicho entorno. Todos los archivos contenidos en el CD del proyecto, están basados en la versión Linux Ubuntu, de manera que si realiza la instalación sobre otra distribución como Suse, Mandrake..., se deberán buscar los archivos en Internet o en el repositorio antes de instalar nada.

Tanto para el modo estándar como para el personalizado, es necesario instalar las siguientes aplicaciones.



- Java Runtime Enviroment (JRE).
- MySQL.
- Apache Tomcat.

Además para el modo personalizado, es conveniente la instalación del entorno de desarrollo Eclipse.

A continuación se explica la instalación de los componentes software necesarios.

#### **7.4.1.- JAVA RUNTIME ENVIROMENT 6 UPDATE 17 (JRE)**

---

JRE es un conjunto de utilidades que permiten la ejecución de programas Java.

Para llevar a acabo su instalación, podemos instalar el archivo contenido en el CD del proyecto en la ruta “Linux/Programas/jre-6u17-linux-i586.bin” que incluye la versión 6 update 17, si lo deseamos podemos descargar dicho archivo de la siguiente dirección:

*<http://java.sun.com/javase/downloads/index.jsp>*

Para su instalación se recomienda seguir los manuales que pueden encontrarse en Internet, en caso de ser necesario.

Otra opción y la más sencilla, es instalar JRE desde el repositorio del sistema, a través del siguiente comando:

*`sudo apt-get install sun-java6-jre`*

#### **7.4.2.- MYSQL 5.1**

---

MySQL es el gestor de bases de datos utilizado en la aplicación, para su instalación, podemos descomprimir el archivo contenido en el CD del proyecto en la ruta “Linux/Programas/mysql-5.1.41-linux-i686-glibc23.tar.gz” o si lo deseamos podemos descargar dicho archivo de la siguiente dirección:

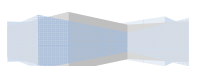
*<http://dev.mysql.com/downloads/mysql/5.1.html#linux>*

Para continuar con la instalación desde el archivo, se recomienda leer el archivo “readme” que explica detalladamente como instalar la aplicación.

Otra opción y la más sencilla, es instalar MySQL desde el repositorio del sistema, a través del siguiente comando, que además instala la consola gráfica de administración de MySQL:

*`sudo apt-get install mysql-server-5.1 mysql-admin`*

Después de ejecutar el comando deberá introducir la contraseña de MySQL, se recomienda que sea “yankoa” a no ser que después vaya a realizar la ejecución





personalizada, en cuyo caso puede elegirse cualquier contraseña. Además, se debe elegir el tipo de configuración que es “intacta”.

### 7.4.3.- APACHE TOMCAT 5.5

Apache Tomcat es el servidor de aplicaciones utilizado para poner en marcha la aplicación, para su instalación, podemos descomprimir el archivo contenido en el CD del proyecto en la ruta “Linux/Programas/apache-tomcat-5.5.28.tar.gz” o si lo deseamos podemos descargar la distribución Core de la siguiente dirección:

*<http://tomcat.apache.org/download-55.cgi>*

Para continuar con la instalación desde el archivo, se recomienda leer el archivo “running” que explica detalladamente como instalar la aplicación. Además y en caso de seguir la ejecución estándar, Tomcat debe encontrarse instalado en los siguientes directorios:

➤ /etc/tomcat5.5

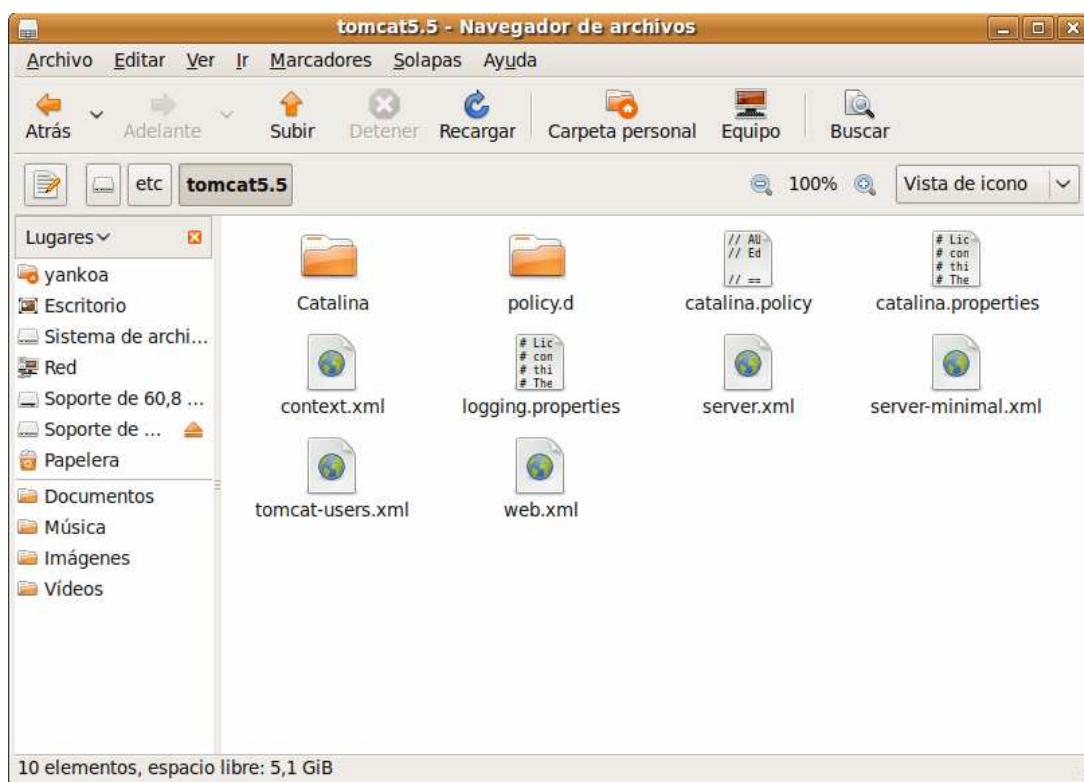
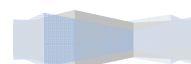


Figura 34: Contenido de /etc/tomcat5.5

➤ /usr/share/tomcat5.5



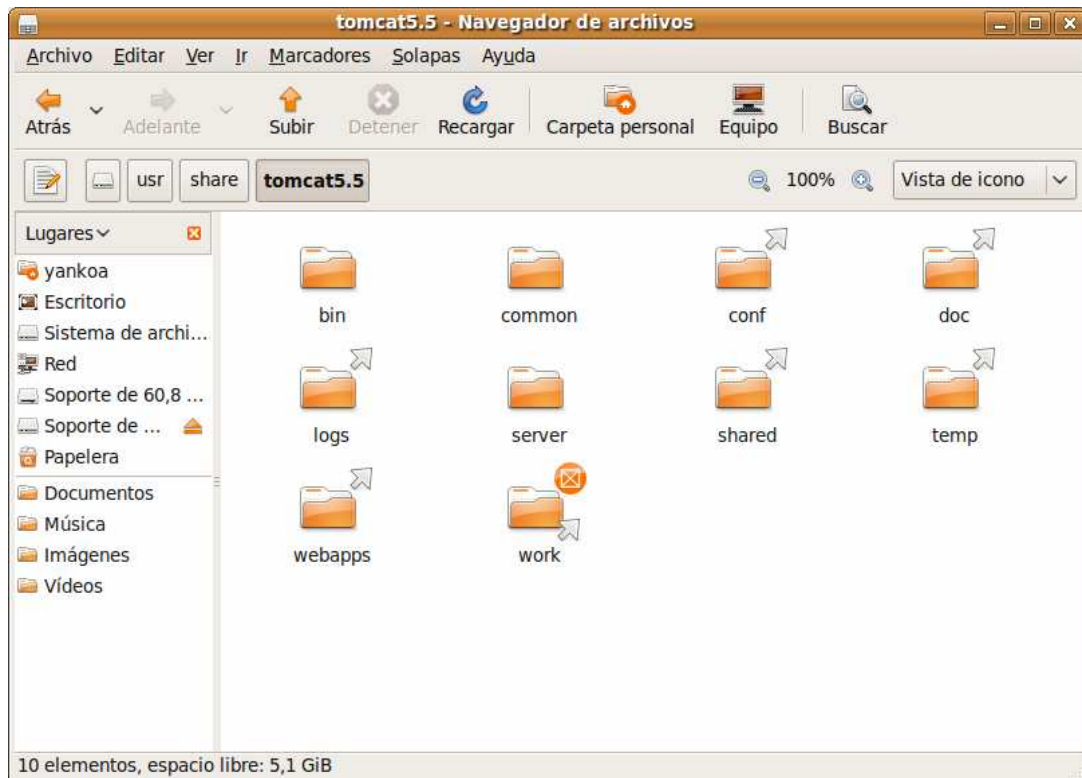


Figura 35: Contenido de /usr/share/tomcat5.5

Otra opción y la más sencilla, es instalar Apache Tomcat desde el repositorio del sistema, a través del siguiente comando:

```
sudo apt-get install tomcat5.5
```

## 7.5.- EJECUCIÓN EN MODO ESTÁNDAR

Tras la instalación de los programas necesarios, es posible ejecutar el programa en modo estándar, es decir, mediante la configuración por defecto.

Para ello, deben seguirse los siguientes pasos:

- 1) Importar la base de datos en MySQL. Primero debemos activar MySQL introduciendo por consola el siguiente comando:

```
sudo /etc/init.d/mysql start
```

Para importar nuestra BD, tenemos dos opciones:

- a. Abrir el administrador de MySQL mediante el siguiente comando:

```
mysql-admin &
```

En la primera pantalla, debemos configurar la conexión, de la siguiente manera.

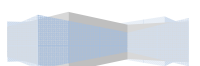




Figura 36: MySQL admin I

Tras ello se abrirá el administrador.

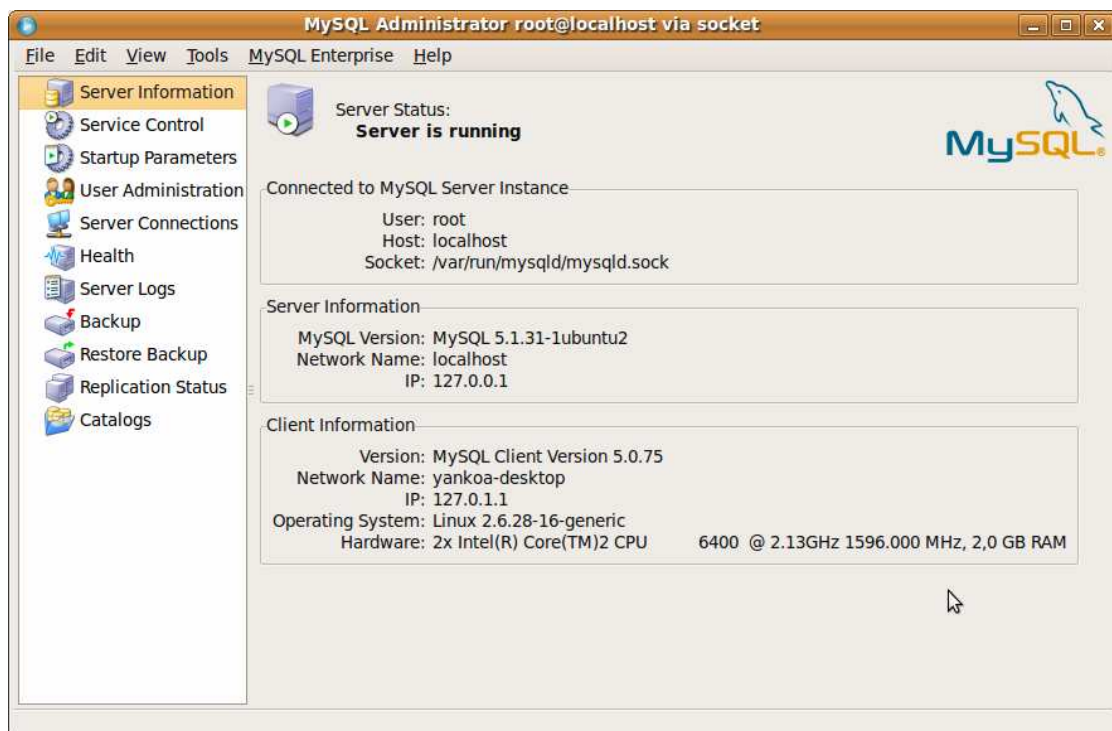
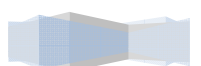


Figura 37: MySQL admin II

Pulsaremos sobre la opción “Restore Backup” apareciéndonos la siguiente pantalla.



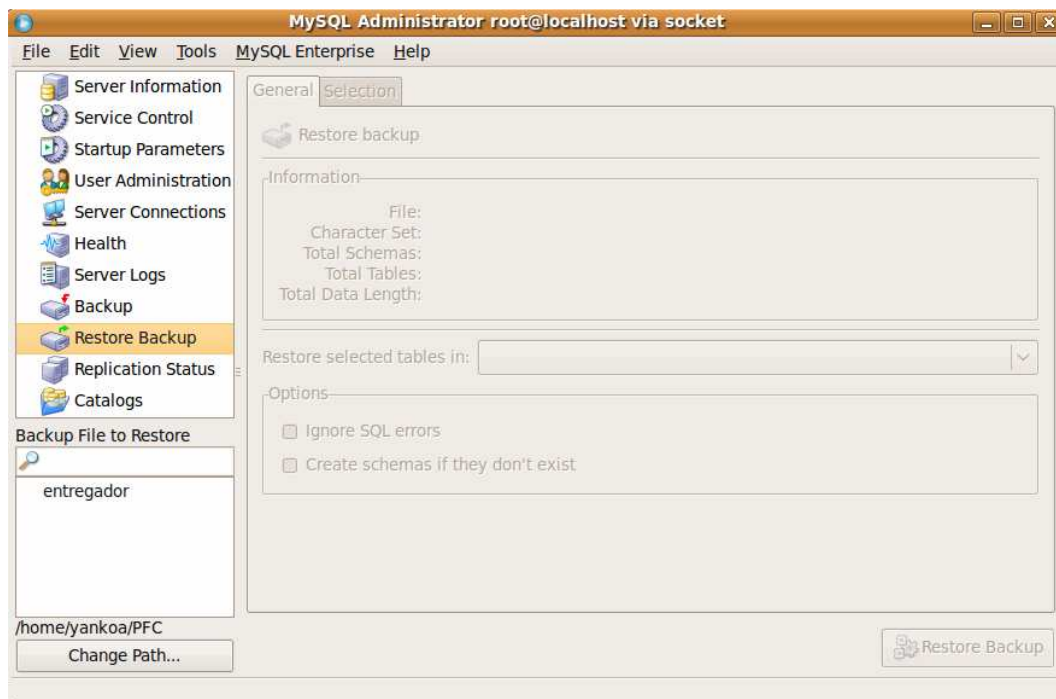


Figura 38: MySQL admin III

En ella debemos buscar en el path que deseemos la BD a cargar, la cual se encuentra en el CD del proyecto en la siguiente ruta “BD/Entregador.sql”. Después nos indicará el formato de texto, el cual dejaremos intacto.

Finalmente pulsaremos sobre el botón “Restore backup” y ya tendremos la BD importada.

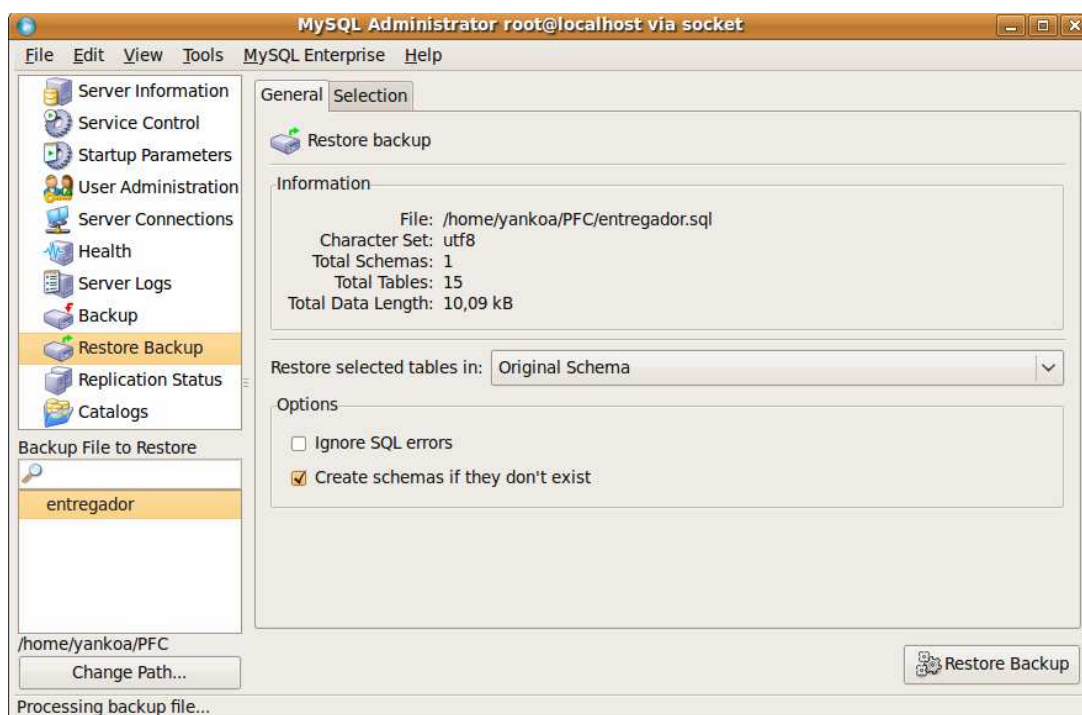
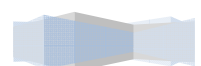


Figura 39: MySQL admin VI



- b. La siguiente opción es importar la BD desde la línea de comandos, introduciendo los siguientes comandos por consola.

```
mysql -u root -p
Create database entregador
source /home/yankoa/entregador.sql
```

- 2) Se debe colocar el conector de la BD dentro de las librerías de Tomcat, para que de esta forma se pueda realizar la conexión con la BD cargada anteriormente. Debemos seguir los siguientes pasos:

- a. Descomprimir el archivo *mysql-connector-java-5.1.8* contenido en el CD del proyecto en la carpeta “BD”.
- b. Copiar el archivo *mysql-connector-java-5.1.8-bin.jar* en las librerías de Tomcat, en la siguiente ruta “/usr/share/Tomcat 5.5/common/lib”.

- 3) Generar el almacén de claves o utilizar el existente.

- a. Si desea generar un almacén de claves, debe abrir una consola de comandos y colocarse en la ruta donde se encuentra instalada su máquina virtual. Tras ello, debe teclear el siguiente comando:

```
keytool -genkey -alias tomcat -keyalg RSA
```

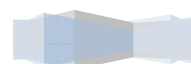
Después debe rellenar todos los datos necesarios, y utilizar como contraseña la palabra “changeit”. Tras ello, se generará un fichero de claves con el nombre “.keystore” en la siguiente ruta “/home/MiNombreDeUsuario”.

- b. Tanto si ha generado el fichero de claves, como si desea hacer uso del que se encuentra en el CD del proyecto en la siguiente ruta “*Almacen claves/.keystore*”, debe colocar dicho fichero en la siguiente ruta, para conseguir que el programa funcione correctamente “/usr/share/Tomcat 5.5”
- 4) Configurar el fichero del servidor Tomcat (server.xml) que se encuentra en la ruta “/usr/share/Tomcat 5.5/conf”, descomentando la parte donde se crea un servidor seguro, es decir, SSL.

- a. La parte a descomentar es la siguiente:

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->
<Connector port="8443" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25"
    maxSpareThreads="75" enableLookups="false"
    disableUploadTimeout="true"
    acceptCount="100" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS" />
```

- b. Al final de ella, debemos añadir el lugar donde se encuentra el fichero de claves y la contraseña. Mediante la siguiente línea:



```
keystoreFile="/usr/share/Tomcat 5.5/.keystore"  
keystorePass="changeit"/>
```

- 5) Colocar el archivo con extensión .war que se encuentra en el CD del proyecto “Linux/War/Entregador.war” dentro de la carpeta webapps de Tomcat, que debe encontrarse en la siguiente ruta “/usr/share/Tomcat 5.5”. De esta manera, el servidor de aplicaciones, logrará desplegarla para su ejecución.
- 6) Una vez hechos todos los pasos, debemos comenzar el servicio de Tomcat mediante el siguiente comando

*sudo /etc/init.d/tomcat5.5 start*

Se destaca que todos los archivos del servidor, se almacenarán en el siguiente directorio “/usr/share/Tomcat 5.5/Entregador”, en caso de no desear que se almacenen en esa ruta, deberá seguir las instrucciones para ejecutar en modo personalizado.

## 7.6.- EJECUCIÓN EN MODO PERSONALIZADO

Tras la instalación de los programas, es posible ejecutar la aplicación de un modo personalizado, esto quiere decir, que deberá crear el fichero .war una vez se hayan configurado todas las rutas por defecto.

Para modificar las rutas por defecto, basta con abrir los ficheros donde se encuentran los valores a modificar, una opción es hacerlo manualmente desde cualquier editor de texto, aunque es mas sencillo desde el entorno de desarrollo integrado Eclipse.

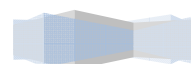
En primer lugar, puede que al generar la BD, haya indicado una contraseña diferente a la indicada anteriormente, por lo tanto, deberá modificar la contraseña del fichero de configuración de Hibernate para que la aplicación se ejecute correctamente. Para ello debe modificar el fichero hibernate.cfg.xml que se encuentra en el CD del proyecto en la ruta “Codigo\Entregador\src” y modificar la siguiente línea cambiando la contraseña:

```
<property name="connection.password">yankoa</property>
```

Tras ello, puede llevar a cabo los pasos 1 y 2 del apartado 7.5.- *Ejecución modo estándar*.

Por otro lado, también puede desear, generar un nuevo almacén de claves siguiendo el paso 3 del apartado 7.5.- *Ejecución modo estándar* y que desee colocarlo en una ruta diferente a la mencionada en ese apartado. Si es así, después de generar el fichero y colocarlo en la ruta deseada, debe abrir el fichero con el nombre Constantes.java que se encuentra en el CD del proyecto en “Codigo\Entregador\src\utilidades” y modificar la siguiente constante, indicando la ruta donde usted haya incluido el almacén de claves:

```
public static final String KEYSTORE = "/usr/share/tomcat5.5/.keystore";
```





Después debe realizar el paso 4 del apartado 7.5.- *Ejecución modo estándar* añadiendo en la línea “keystoreFile” la ruta que ha indicado.

Finalmente, puede ser que desee modificar la ruta donde se almacenan los ficheros generados en el servidor, si es así, debe modificar el archivo Constantes.java que se encuentra en el CD del proyecto en “Codigo\Entregador\src\utilidades” y modificar la siguiente constante, indicando la ruta donde usted desee almacenar todos los datos:

```
public static final String PROYECTO = "Entregador/";
```

Una vez configurados todos los elementos, se debe proceder a generar el fichero .war para poder desplegarlo en el servidor, para ello tenemos dos opciones.

- Si nos encontramos dentro de Eclipse, debemos pulsar con el botón derecho en el proyecto sobre la opción exportar a war. Aquí elegimos el nombre del war, que se recomienda sea Entregador, la ruta donde deseamos almacenarlo además de indicar el servidor Tomcat 5.5.
- Si no deseamos usar Eclipse, se puede generar el war desde la línea de comandos, situándonos en el directorio del proyecto, debemos teclear lo siguiente:

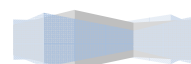
```
jar cvf Entregador.war *
```

Cuando ya hemos generado el war, debemos seguir los pasos 5 y 6 del apartado 7.5.- *Ejecución modo estándar*.

## 7.7.- POSIBLES ERRORES

Siguiendo todos los pasos descritos anteriormente, no debe producirse ningún error, pero a pesar de ello, vamos a indicar los errores más comunes y su forma de solventarlos.

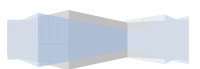
- Se intenta arrancar la aplicación pero aparece un error indicando que no se puede establecer una conexión con el servidor. Es error se produce cuando no se ha arrancado el servidor Apache Tomcat, por lo tanto debe arrancarse como se ha indicado anteriormente.
- Tras intentar logearse en la aplicación o acceder como alumno, aparece un mensaje de error. Esto puede deberse a que la conexión con la BD se ha perdido, en ese caso deberá arrancarla de nuevo si no lo ha hecho antes. Otro problema puede deberse a que la BD no ha sido correctamente configurada, de manera que deberán revisarse todos los parámetros de configuración descritos anteriormente.
- Al realizar una acción aparece un mensaje de error, si se descarta el anterior error, puede ser debido a un error en la pila de la máquina virtual de Java, que se conoce como “Java head space”. Para evitar este error, se debe ampliar el espacio de la zona de intercambio siempre teniendo en cuenta que dicho espacio estará limitado por la cantidad de memoria de la máquina donde se





este ejecutando la aplicación. La solución consiste en introducir el siguiente parámetro en la JVM:

```
java -Xms<tamaño de inicio de la memoria> -Xmx<tamaño máximo>
```





## 8.- ORGANIZACIÓN DEL PROYECTO

### 8.1.- ROLES ORGANIZATIVOS Y RESPONSABILIDADES

Para llevar a cabo este proyecto, he debido actuar siguiendo los siguientes roles, de modo que se consiguiera cubrir en su mayor parte, el equipo de trabajo necesario para este tipo de aplicaciones.

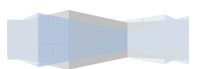
- **Jefe de Proyecto:** Encargado de supervisar todo el proyecto y servir como nexo de unión entre el cliente y el resto de roles. Entre sus principales objetivos se destacan:
  - Establecer la planificación de todo el proyecto, es decir, fechas de inicio y fin de tareas, reuniones con el cliente...
  - Realizar seguimiento exhaustivo del proyecto en todo momento, cerciorándose de que todo estuviera en orden.
  - Asignar los recursos necesarios, tanto materiales como inmateriales, a cada fase del proyecto.
- **Analista:** Encargado de las siguientes tareas:
  - Conocer en detalle los requerimientos del sistema y entenderlos.
  - Realizar el documento de requisitos y la especificación de los mismos para validarlos con el cliente.
  - Verificar la trazabilidad entre los distintos modelos.
- **Diseñador:** Desempeñando las siguientes tareas:
  - Transformar los requerimientos en el diseño de lo que debe ser el sistema.
  - Desarrollar una arquitectura robusta para el sistema.
  - Adaptar el diseño para que se corresponda con el entorno de implementación, diseñando de forma que ésta tenga una correcta ejecución.
- **Programador:** Las tareas fundamentales han sido la codificación del software de acuerdo al documento de diseño detallado y la ejecución de las pruebas para comprobar su correcto funcionamiento.





## 8.2.- PLANIFICACIÓN

A continuación, se muestra la planificación del proyecto a través de un diagrama Gantt, en el cual se reflejan cada una de las tareas realizadas en el proyecto, así como su duración y los recursos asignados.



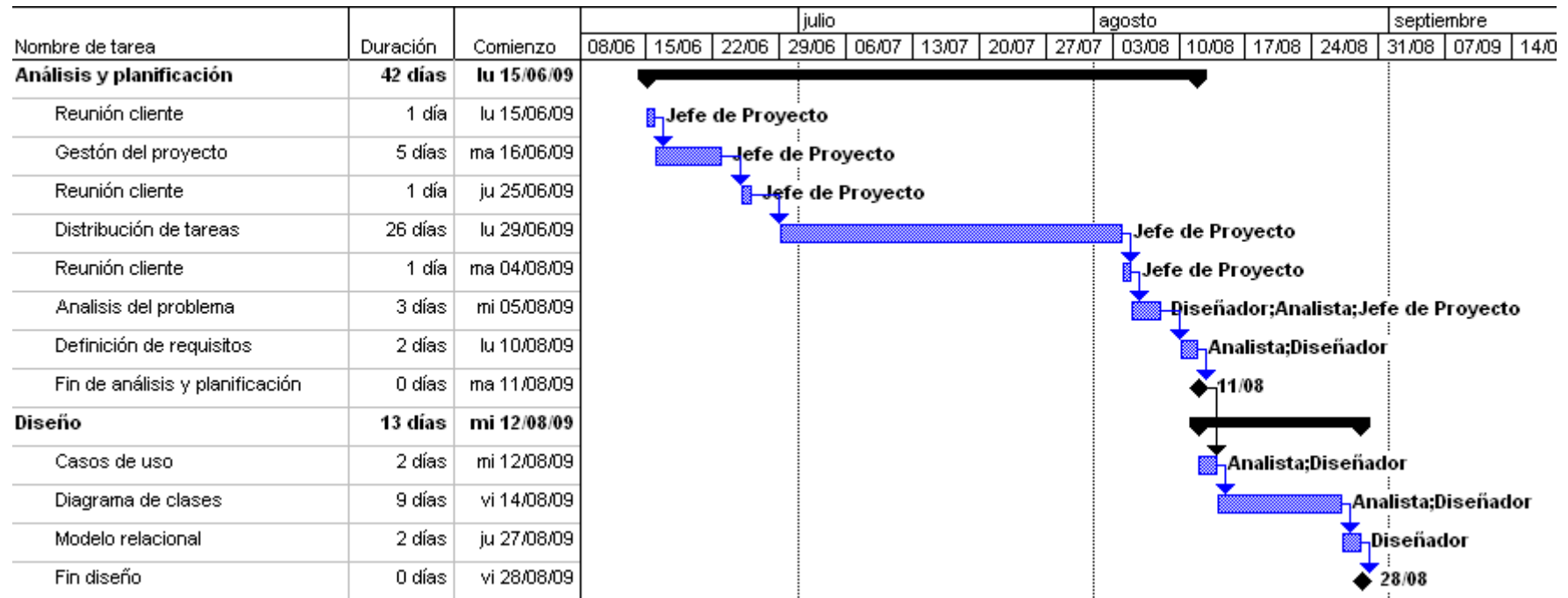
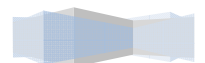


Figura 40: Diagrama Gantt I



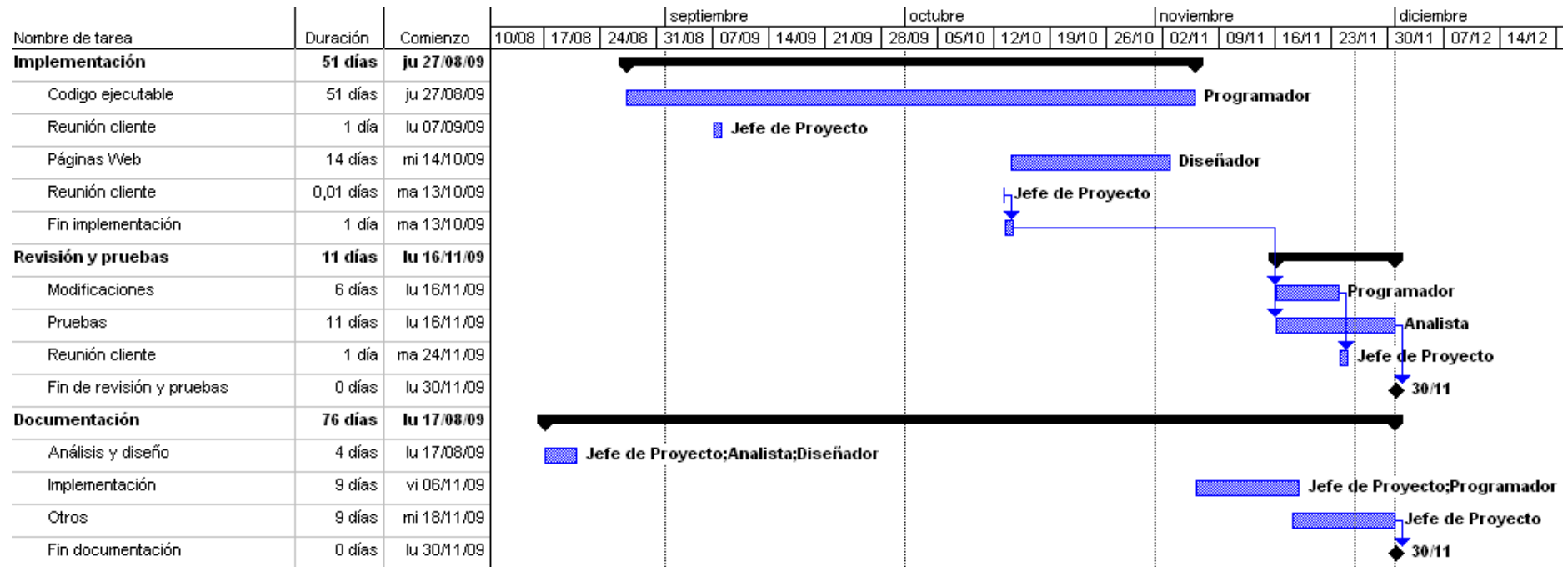


Figura 41: Diagrama Gantt II



### 8.3.- PRESUPUESTO

A continuación se muestran los costes y gastos incurridos para la realización del proyecto.

#### 8.3.1.- COTES DE PERSONAL

Se consideran costes de personal todos aquellos gastos económicos del proyecto relacionados con los recursos humanos necesarios para llevarlo a cabo.

El equipo del proyecto se compone de 4 roles cuyos costes asociados con los siguientes:

ROL	SUELDO ANUAL	SUELDO MENSUAL	SUELDO DÍA	SUELDO HORA
Jefe de Proyecto	37.000 €	3.083 €	140 €	18 €
Analista	27.000 €	2.250 €	102 €	13 €
Diseñador	24.000 €	2000 €	91 €	11 €
Programador	21.000 €	1.750 €	80 €	10 €

Tabla 12: Costes de personal I

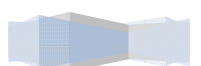
Una vez especificados los costes hora de cada miembro del equipo, se detallarán los costes totales de cada miembro del equipo, según las horas que han trabajado.

ROL	SUELDO HORA	HORAS TRABAJO	COSTE TOTAL
Jefe de Proyecto	18 €	73	1.314 €
Analista	13 €	49	637 €
Diseñador	11 €	20	220 €
Programador	10 €	140	1400 €
<b>TOTAL</b>		<b>282</b>	<b>3.571 €</b>

Tabla 13: Costes de personal II

#### 8.3.2.- COTES DE FORMACIÓN

Ha sido necesario formar a los miembros del grupo de trabajo en algunas de las tecnologías usadas, especialmente Struts, Hibernate y entornos Linux.



CONCEPTO	COSTE HORA	HORAS CURSO	COSTE TOTAL
Curso Struts	12 €	10	120 €
Curso Hibernate	12 €	5	75 €
Curso Linux	12 €	12	144 €
<b>TOTAL</b>		<b>27</b>	<b>339 €</b>

Tabla 14: Costes de formación

### 8.3.3.- COTES DE MATERIAL

En este punto se detallarán los costes asociados al hardware, software y al material fungible asociados al proyecto.

En cuanto a material hardware, han sido necesarios el uso de diversos equipos informáticos para el desarrollo del proyecto, además de unidades extraíbles de datos para el almacenamiento seguro de los mismos y por supuesto impresoras para todos los documentos necesarios.

CONCEPTO	UNIDADES	COSTE TOTAL
PC sobremesa Intel Core 2 Duo 2.13gh	2	2.000 €
Portátil Dell Precision™ M4400	1	1.000 €
Impresora HP PhotoSmart C4280	1	80 €
Dispositivo almacenamiento USB Kingston DataTraveler Memoria USB 2 GB	1	15 €
Dispositivo almacenamiento disco duro extraíble 40 GB	1	40 €
<b>TOTAL</b>		<b>3.135 €</b>

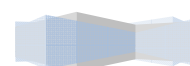
Tabla 15: Costes de hardware

En cuanto al software, todos los programas utilizados han sido de código abierto y por lo tanto, no se ha incurrido ningún gasto derivado de su uso.

Como material fungible, se han contabilizado elementos materiales como fotocopias y material de oficina.

CONCEPTO	UNIDADES	COSTE TOTAL
Material de oficina		10 €
Cartucho negro de inyección de tinta HP 350	1	16,90 €
Cartucho de inyección de tinta tricolor HP 350	1	19,91 €
CD-R Verbatim 52x	25	6,90 €
<b>TOTAL</b>		<b>61,8 €</b>

Tabla 16: Costes de material fungible



Finalmente se han contabilizado otros gastos, que también han tenido lugar en el proyecto, como son el transporte para las reuniones con el cliente y la conexión a Internet.

CONCEPTO	COSTE TOTAL
Desplazamientos	7,40 €
Conexión a Internet banda ancha 50Mb Ono	250 €
<b>TOTAL</b>	<b>257,40 €</b>

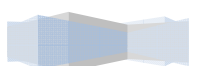
Tabla 17: Otros costes

### 8.3.4.- COTE TOTAL

A modo de resumen se refleja el coste total del proyecto, a través de los costes calculados anteriormente.

DESCRIPCIÓN	COSTE TOTAL
Personal	3.571 €
Formación	339 €
Hardware	3.135 €
Software	0 €
Material fungible	61,8 €
Otros gastos	257,4 €
<b>TOTAL</b>	<b>7.364,2 €</b>

Tabla 18: Coste total del proyecto



## 9.- CONCLUSIONES

El resultado obtenido tras la realización del proyecto ha sido una aplicación Web en la que se permiten multitud de funciones a realizar por los usuarios y que esta implementada a través de novedosas tecnologías que la dotan de amplias posibilidades de reutilización y modificación. Por lo tanto, el proyecto ha cumplido con todos los objetivos marcados.

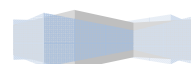
Gracias a mi experiencia en otros trabajos similares utilizando las mismas tecnologías que para este proyecto, no he encontrado grandes dificultades en su implementación. A pesar de ello, siempre surgen pequeños problemas, especialmente respecto a elementos que nunca había implementado como por ejemplo la compresión y descompresión de ficheros con Java y la parte de certificados digitales. En estos aspectos he tenido que buscar mucha información y realizar diversas pruebas antes de lograr que todo funcionara.

Una de las partes más complicadas ha sido la interacción de la aplicación tanto en Explorer como en Mozilla, ya que las antiguas versiones de Explorer no cumplían los estándares mínimos de seguridad de la W3C y por lo tanto resultaba muy difícil conseguir que las páginas se visualizaran correctamente en ambos exploradores y que su funcionamiento fuera el mismo. Esto dio lugar a tomar la decisión de utilizar el navegador Explorer en su última versión, ya que este si esta adaptado.

Me gustaría destacar que las partes mas importantes del proyecto son el uso de los patrones Struts e Hibernate, ya que gracias a ellos, se logra un código mucho más limpio, la futura manutención del mismo en caso de cambios y una separación muy clara de todas las partes especialmente del diseño y la implementación.

En cuanto a posibles líneas futuras, el proyecto puede ser ampliado y mejorado, por ejemplo en los siguientes aspectos:

- **Correcciones automáticas:** Conseguir que la aplicación a través de alguna otra pueda corregir las prácticas de forma automática cada cierto periodo de tiempo, generando así las calificaciones de las mismas.
- **Comprobación de copias:** Permitir que la aplicación compruebe si las prácticas entregadas son copiadas o no, todo ello también a través de un programa externo que se una a la aplicación actual.
- **Registro o log de los usuarios:** Intentar registrar todos lo que hacen los usuarios cuando están conectados a la aplicación, con el fin de poder comprobar si están cometiendo algún error. Esta parte se ha intentado implementar, pero el problema es que la BD podía crecer de una manera descomunal y provocar graves errores, por lo que es una buena parte a estudiar para futuras ampliaciones.





## 10.- BIBLIOGRAFÍA

En este apartado se recogen todos los documentos y páginas Web utilizadas para la realización del proyecto.

**Título:** Aprenda Java como si estuviera en primero.

**Autor:** Universidad de Navarra.

**Título:** Jakarta Struts.

**Autor:** Chuck Cavaness.

**Editorial:** Anaya.

**Título:** Compresión y descompresión de datos utilizando Java.

**Disponible en:** [http://www.javahispano.org/contenidos/es/compresion\\_en\\_java/](http://www.javahispano.org/contenidos/es/compresion_en_java/)

**Título:** Hibernate in action.

**Autor:** Christian Bayer y Gavin King.

**Editorial:** Hanning.

**Título:** Hibernate reference documentation version 3.2.2.

**Autor:** Hibernate.

**Título:** Análisis comparativo de bases de datos de código abierto vs código cerrado.

**Autor:** Diego Javier Burbano Proaño.

**Disponible en:** <http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf>

**Título:** API Java Enterprise Edition 6.

**Disponible en:** <http://java.sun.com/javaee/6/docs/api/>

**Título:** Firma digital.

**Disponible en:** <http://java.sun.com/docs/books/tutorial/security/apisign/index.html>

**Título:** Foros del Web.

**Disponible en:** <http://www.forosdelweb.com/>

**Título:** Página de referencia de Hibernate.

**Disponible en:** <https://www.hibernate.org>

**Título:** Página de referencia de MySQL.

**Disponible en:** <http://www.mysql.com>

**Título:** Página de referencia de Struts.

**Disponible en:** <http://struts.apache.org>





**Título:** Página para activar un certificado digital en Apache Tomcat.

**Disponible en:** <http://www.locualo.net/programacion/activar-ssl-tomcat-certificado-digital-prueba/00000081.aspx>

**Título:** Wikipedia

**Disponible en:** <http://es.wikipedia.org/wiki/Wikipedia:Portada>



## ANEXO I.- MANUALES DE USUARIO

En este apartado se describirá el uso de la aplicación para cada uno de los usuarios del sistema.

Para ello, se mostrarán 3 manuales en los que diferenciaremos 4 roles:

- Administrador.
- Coordinadores y profesores.
- Alumnos.

### 1.- MANUAL DE USUARIO DEL ADMINISTRADOR

El administrador del sistema podrá acceder a la aplicación introduciendo en cualquier navegador Web (se recomienda Internet Explorer 8 o Mozilla Firefox 3 o superior) la siguiente dirección:

*<https://<host>:<puerto>/Entregador/>*

Donde “host” se corresponderá con el nombre de una máquina y “puerto”, el puerto por el que correrá la aplicación.

Tras ello aparecerá la pantalla de acceso.



Universidad  
Carlos III de Madrid

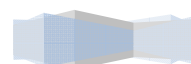
Entregador de prácticas



Figura 42: Acceso al sistema

En ella, el administrador debe introducir los siguientes datos:

- Identificador de usuario: admin.
- Contraseña: Debe contener al menos 7 caracteres y se recomienda la inclusión de números y signos de puntuación para hacerla más segura. La primera vez que acceda deberá introducir como contraseña “1234567” (se recomienda cambiarla).



Tras completar los datos, se debe enviar el formulario pulsando sobre el botón “Entrar”. En caso de producirse cualquier error, se informará al administrador con el fin de que vuelvan a introducir los datos correctamente.

Si los datos han sido introducidos correctamente, aparecerá la siguiente pantalla:



Figura 43: Inicio administrador

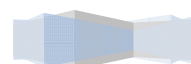
A partir de este momento el administrador se encontrará identificado y podrá acceder a cualquiera de las opciones del menú:

- Administrador – Modificar contraseña.
- Personal docente – Activar.
- Personal docente – Buscar/Modificar.
- Coordinadores – Activar.
- Coordinadores – Buscar/Modificar/Borrar.
- Cerrar sesión.

A continuación, se explicarán cada una de las opciones del menú.

### 1.1.- ADMINISTRADOR – MODIFICAR CONTRASEÑA

Se recomienda que el administrador modifique su contraseña tras acceder por primera vez al sistema. A pesar de ello, también es posible modificarla siempre que lo desee.



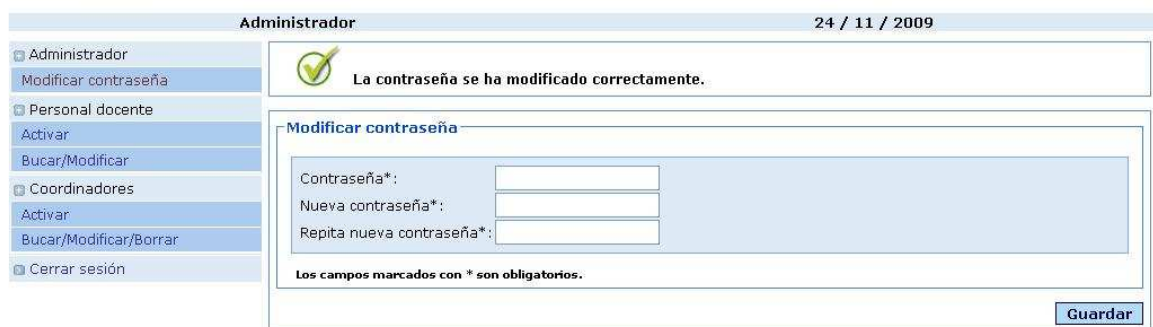


The screenshot shows the 'Administrador' interface with a sidebar menu on the left containing options like 'Administrador', 'Modificar contraseña', 'Personal docente', 'Activar', 'Bucar/Modificar', 'Coordinadores', 'Activar', 'Bucar/Modificar/Borrar', and 'Cerrar sesión'. The main content area is titled 'Modificar contraseña' and contains three input fields: 'Contraseña\*', 'Nueva contraseña\*', and 'Repita nueva contraseña\*'. Below the fields is a note: 'Los campos marcados con \* son obligatorios.' and a 'Guardar' button at the bottom right. The top header shows 'Administrador' and the date '24 / 11 / 2009'.

Figura 44: Modificar contraseña

Para modificar la contraseña el administrador deberá introducir su actual contraseña, la nueva contraseña y repetir la nueva contraseña.

Tras ello deberá enviar el formulario pulsando sobre el botón “Guardar”, si existe algún error el administrador será informado para que modifique los datos introducidos, en caso contrario aparecerá la siguiente pantalla:



The screenshot shows the 'Administrador' interface after a successful password change. A green checkmark icon and the message 'La contraseña se ha modificado correctamente.' are displayed at the top. Below this is the 'Modificar contraseña' form, which is identical to the one in Figure 44, with the same sidebar menu, input fields, and 'Guardar' button. The top header shows 'Administrador' and the date '24 / 11 / 2009'.

Figura 45: Contraseña modificada

## 1.2.- PERSONAL DOCENTE - ACTIVAR

El administrador será el encargado de activar al personal docente de la universidad dentro de la aplicación. Tras pulsar esta opción del menú se mostrará la siguiente pantalla:

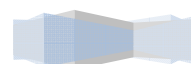




Figura 46: Activar docente

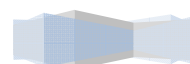
Para activar un docente, se deben introducir los siguientes datos:

- Identificador de usuario: Cadena con un tamaño máximo de 5 caracteres que debe comenzar por una letra. No pueden existir dos docentes con el mismo identificador.
- Contraseña: Contraseña de acceso para el docente. Debe contener al menos 7 caracteres y se recomienda la inclusión de números y signos de puntuación para hacerla más segura.
- Repita contraseña: Se debe escribir una segunda vez la contraseña para evitar errores.
- NIA: Identificador del docente. Debe comenzar por 100 y tener 9 dígitos.
- Nombre.
- Apellidos.
- Email.

Tras introducir los datos, el administrador deberá enviar el formulario pulsando sobre el botón “Guardar”, si existe algún error el administrador será informado para que modifique los datos introducidos, en caso contrario aparecerá la siguiente pantalla:



Figura 47: Docente activado



Desde esta pantalla, el administrador podrá modificar todos los datos personales del docente a excepción del identificador de usuario y la contraseña, pulsando nuevamente en el botón “Guardar”.

### 1.3.- PERSONAL DOCENTE – BUSCAR/MODIFICAR

El administrador puede consultar todos los docentes que ha activado y a partir de ahí modificar sus datos.



Universidad Carlos III de Madrid

Entregador de prácticas

Administrador 24 / 11 / 2009

Administrador

- Administrador
- Modificar contraseña
- Personal docente
  - Activar
  - Buscar/Modificar
- Coordinadores
  - Activar
  - Buscar/Modificar/Borrar
- Cerrar sesión

Buscar docentes

NIA:

Nombre:

Apellidos:

Buscar Mostrar todos

Figura 48: Buscar docente I

Para consultar los docentes, existen dos opciones:

- Pulsar sobre el botón “Mostrar todos”, de modo que se muestren todos los docentes activados.



Universidad Carlos III de Madrid

Entregador de prácticas

Administrador 24 / 11 / 2009

Administrador

- Administrador
- Modificar contraseña
- Personal docente
  - Activar
  - Buscar/Modificar
- Coordinadores
  - Activar
  - Buscar/Modificar/Borrar
- Cerrar sesión

Buscar docentes

NIA:

Nombre:

Apellidos:

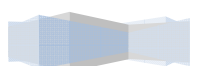
Buscar Mostrar todos

Resultados

Listado de docentes

NIA	Nombre	Apellidos	
100048391	Yankoa	López Cozas	
100048392	Pedro	Ayala Sánchez	
100048393	Ana	López Cozas	
100048394	Prueba	Prueba Prueba	

Figura 49: Buscar docente II





- Escribir el NIA, nombre y/o apellidos del docente/s que desee buscar y pulsar sobre el botón “Buscar”.



Administrador 24 / 11 / 2009

Buscar docentes

NIA:   
Nombre:   
Apellidos:

Buscar Mostrar todos

Resultados

Listado de docentes

NIA	Nombre	Apellidos	
100048394	Prueba	Prueba Prueba	

Figura 50: Buscar docente III

En ambas búsquedas, si no existen resultado se mostrará por pantalla el mensaje “No se han encontrados coincidencias”.

Tras la búsqueda, el administrador puede pulsar sobre el icono de lupa que hay al lado de cada docente y consultar así sus datos y/o modificarlos.

## 1.4.- COORDINADORES - ACTIVAR

El administrador es el encargado de activar a los coordinadores de cada asignatura. Tras pulsar sobre esta opción del menú aparecerá la siguiente pantalla:



Administrador 24 / 11 / 2009

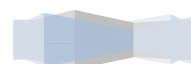
Datos del coordinador

Docente\*:   
Titulación/Plan de estudios\*:   
Año académico (22-22)\*:

Los campos marcados con \* son obligatorios.

Guardar

Figura 51: Activar coordinador





El administrador deberá indicar los siguientes datos:

- Docente: Se debe seleccionar de un campo desplegable el nombre del docente que se desea activar como coordinador.
- Titulación/Plan de estudios: Se debe seleccionar de un desplegable el nombre de la titulación y el plan de estudios al que pertenece la asignatura a coordinar. Tras ello, se cargará otro campo desplegable en el que debe seleccionarse la asignatura y el campus.
- Año académico: Se debe indicar el año académico en el que se coordina la asignatura en formato “aa-aa”, es decir, 09-10.

Tras introducir los datos, el administrador deberá enviar el formulario pulsando sobre el botón “Guardar”, si existe algún error el administrador será informado para que modifique los datos introducidos, en caso contrario aparecerá la siguiente pantalla:



The screenshot shows the 'Entregador de prácticas' interface for the 'Universidad Carlos III de Madrid'. The user is logged in as 'Administrador' on '24 / 11 / 2009'. A success message states 'Los datos se han guardado correctamente.' Below this is the 'Datos del coordinador' form with the following fields:

Docente*:	Prueba Prueba Prueba
Titulación/Plan de estudios*:	Ingeniería Informática - Plan 2008
Asignatura/Campus*:	Redes - Colmenarejo
Año académico (aa-aa)*:	09-10

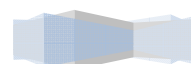
Below the form, it says 'Los campos marcados con \* son obligatorios.' At the bottom right are buttons for 'Guardar' and 'Eliminar'.

Figura 52: Coordinador activado

Desde esta pantalla, el administrador podrá modificar los datos del coordinador a excepción del docente, pulsando nuevamente en el botón “Guardar”. Además podrá eliminar la relación de coordinación entre un docente y una asignatura pulsando sobre el botón “Eliminar”.

## 1.5.- COORDINADORES – BUSCAR/MODIFICAR/BORRAR

El administrador puede consultar todos los coordinadores que ha activado y a partir de ahí modificar sus datos y/o eliminarlos del sistema.



Administrador 24 / 11 / 2009

- Administrador
- Modificar contraseña
- Personal docente
- Activar
- Bucar/Modificar
- Coordinadores
- Activar
- Bucar/Modificar/Borrar
- Cerrar sesión

**Buscar coordinadores**

NIA:  Nombre:   
 Apellidos:  Año académico (aa-aa):   
 Titulación/Plan de estudios: Seleccione titulación/plan

Figura 53: Buscar coordinador I

Para consultar los coordinadores, existen dos opciones:

- Pulsar sobre el botón “Mostrar todos”, de modo que se muestren todos los coordinadores activados.

Administrador 24 / 11 / 2009

- Administrador
- Modificar contraseña
- Personal docente
- Activar
- Bucar/Modificar
- Coordinadores
- Activar
- Bucar/Modificar/Borrar
- Cerrar sesión

**Buscar coordinadores**

NIA:  Nombre:   
 Apellidos:  Año académico (aa-aa):   
 Titulación/Plan de estudios: Seleccione titulación/plan

**Resultados**

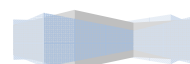
**Listado de coordinadores**

NIA	Nombre	Apellidos	
100048391	Yankoa	López Cozas	
100048392	Pedro	Ayala Sánchez	
100048393	Ana	López Cozas	
100048394	Prueba	Prueba Prueba	

Figura 54: Buscar coordinador II

- Escribir el NIA, nombre, apellidos, año académico, titulación/plan de estudios y/o asignatura/campus del docente/s que desee buscar y pulsar sobre el botón “Buscar”.

En ambas búsquedas, si no existen resultados se mostrará por pantalla el mensaje “No se han encontrados coincidencias”.



Tras la búsqueda, el administrador puede pulsar sobre el icono de lupa que hay al lado de cada coordinador y consultar así sus datos, modificarlos o eliminar al coordinador.

## 1.6.- CERRAR SESIÓN

Tras finalizar de realizar acciones, se recomienda al administrador que pulse sobre esta opción para cerrar su sesión y evitar que otros usuarios puedan acceder a ella y modificar sus datos o realizar acciones indebidas.

## 2.- MANUAL DE USUARIO DE COORDINADORES Y PROFESORES

Los coordinadores y profesores de las asignaturas podrá acceder a la aplicación introduciendo en cualquier navegador Web (se recomienda Internet Explorer 8 o Mozilla Firefox 3 o superior) la siguiente dirección:

*https://<host>:<puerto>/Entregador/*

Donde “host” se corresponderá con el nombre de una máquina y “puerto”, el puerto por el que correrá la aplicación.

Tras ello aparecerá la pantalla de acceso.



Universidad  
Carlos III de Madrid

Entregador de prácticas


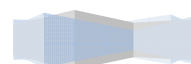


Figura 55: Acceso al sistema

En ella, deberán introducir los siguientes datos:

- Identificador de usuario: Cadena de tamaño máximo 5 caracteres que debe comenzar por una letra.
- Contraseña: Debe contener al menos 7 caracteres y se recomienda la inclusión de números y signos de puntuación para hacerla más segura.

Tras completar los datos, se debe enviar el formulario pulsando sobre el botón “Entrar”. En caso de producirse cualquier error, se informará al usuario con el fin de que vuelvan a introducir los datos correctamente.



Si los datos han sido introducidos correctamente, aparecerá la siguiente pantalla:




Universidad  
Carlos III de Madrid

Entregador de prácticas

Usted coordina y/o imparte diferentes asignaturas. Seleccione el modo en que desea acceder a la aplicación.

**Asignaturas coordinadas**

Listado de asignaturas coordinadas					
	Titulación	Plan estudios	Asignatura	Campus	Año académico
	Ingeniería Informática	Plan 2008	Redes	Colmenarejo	09-10

**Asignaturas impartidas**

No se han encontrado coincidencias.

[Cerrar sesión](#)

Figura 56: Seleccionar asignatura

En ella el docente deberá indicar a través de que asignatura desea acceder a la aplicación y si en modo coordinador o profesor. También tiene la opción de cerrar la sesión.

Las posibles opciones que pueden producirse son:

- Acceso al sistema como coordinador de una asignatura.



Universidad  
Carlos III de Madrid

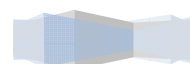
Entregador de prácticas

Coordinador: a0004    Titulación: Ingeniería Informática    Asignatura: Redes    Año: 09-10    24 / 11 / 2009

- ☐ Coordinador
  - Modificar contraseña
  - Cambiar de asignatura
- ☐ Profesores
  - Activar
  - Bucar/Modificar/Borrar
- ☐ Prácticas
  - Alta
  - Bucar/Modificar/Borrar
  - Obtener URL
  - Recoger prácticas
  - Comprobar firma digital
- ☐ Alumnos
  - Importar/Consultar listas
  - Consultar grupos
- ☐ Cerrar sesión



Figura 57: Inicio coordinador



- Acceso al sistema como profesor con permisos de una asignatura.



**Figura 58: Inicio profesor con permisos**

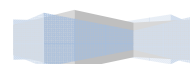
- Acceso al sistema como profesor sin permisos de una asignatura.



**Figura 59: Inicio profesor**

Como se puede observar la diferencia entre cada tipo de acceso es que cada menú tiene diferentes opciones. El menú del coordinador es el que más opciones tiene, mientras que el menú del profesor sin permisos tiene solo algunas de las opciones del coordinador.

A continuación se muestran todas las opciones posibles, que serán realizadas por el personal que tenga acceso a ellas:





- Coordinador/Profesor – Modificar contraseña.
- Coordinador/Profesor – Cambiar asignatura.
- Profesores – Activar.
- Profesores – Buscar/Modificar/Borrar.
- Prácticas – Alta.
- Prácticas – Buscar/Modificar/Borrar.
- Prácticas – Obtener URL.
- Prácticas – Recoger prácticas.
- Prácticas – Comprobar firma digital.
- Alumnos – Importar/Consultar listas.
- Alumnos – Consultar grupos.
- Cerrar sesión.

## 2.1.- COORDINADOR/PROFESOR – MODIFICAR CONTRASEÑA

Tanto los coordinadores como los profesores pueden modificar siempre que lo desee su contraseña de acceso del sistema.

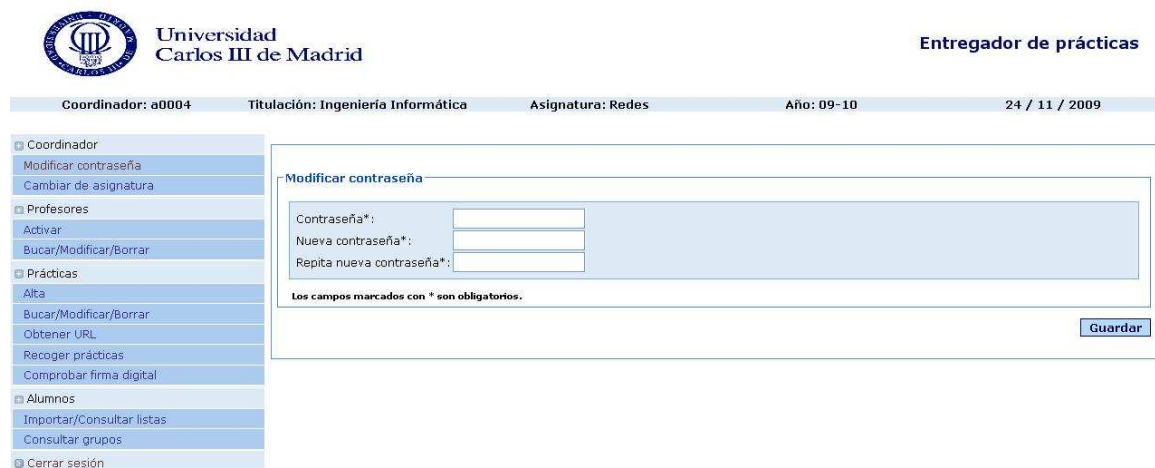


Figura 60: Modificar contraseña

Para modificar la contraseña deberán introducir su actual contraseña, la nueva contraseña y repetir la nueva contraseña.

Tras ello deberá enviar el formulario pulsando sobre el botón “Guardar”, si existe algún error serán informado para que modifiquen los datos introducidos, en caso contrario aparecerá la siguiente pantalla:

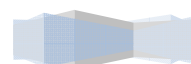




Figura 61: Contraseña modificada

## 2.2.- COORDINADOR/PROFESOR – CAMBIAR ASIGNATURA

En cualquier momento los coordinadores y profesores pueden acceder a la aplicación a través de otra asignatura que coordinen o impartan sin tener que volver a identificarse.

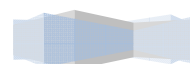
Se mostrará la misma pantalla que en la *figura 56* y deberán elegir la asignatura a través de la cual desean acceder.

## 2.3.- PROFESORES - ACTIVAR

Los coordinadores de cada asignatura serán los encargados de activar a los profesores de dicha asignatura para un curso académico concreto. Tras pulsar esta opción del menú se mostrará la siguiente pantalla:



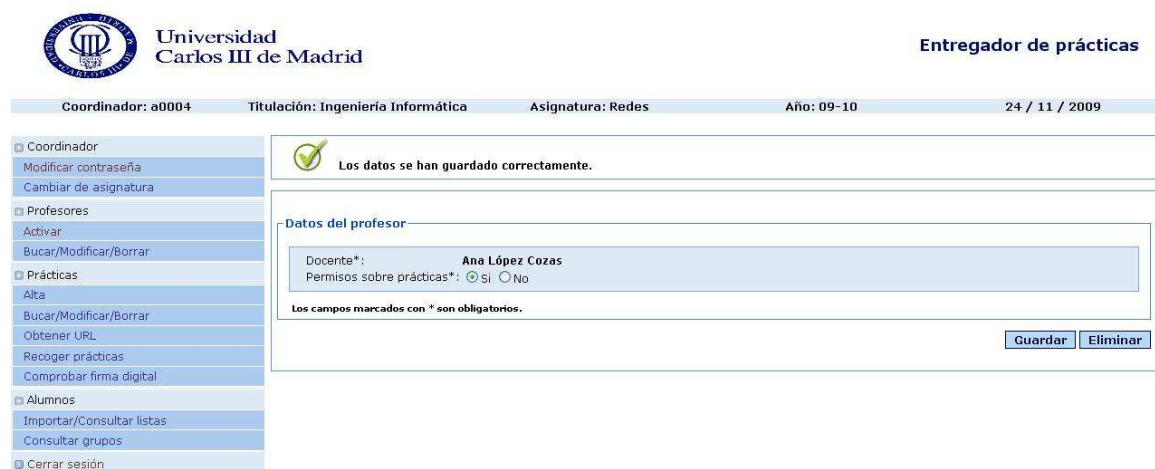
Figura 62: Activar profesor



Para activar un profesor, se deben introducir los siguientes datos:

- Docente: Se debe seleccionar de un campo desplegable el nombre del docente que se desea activar como profesor.
- Permisos sobre prácticas: Se deberá indicar si el profesor tendrá o no permisos sobre las prácticas, es decir, si podrá crearlas, consultarlas, modificar y borrarlas.

Tras introducir los datos, el coordinador deberá enviar el formulario pulsando sobre el botón “Guardar”, si existe algún error el coordinador será informado para que modifique los datos introducidos, en caso contrario aparecerá la siguiente pantalla:



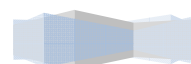
The screenshot shows the 'Entregador de prácticas' interface for Universidad Carlos III de Madrid. The header includes the university logo and name. The navigation menu on the left lists options like 'Coordinador', 'Profesores', 'Prácticas', and 'Alumnos'. The main content area displays a success message: 'Los datos se han guardado correctamente.' Below this is the 'Datos del profesor' form, which contains a dropdown for 'Docente\*' (set to 'Ana López Cozas') and radio buttons for 'Permisos sobre prácticas\*' (set to 'Si'). A note states 'Los campos marcados con \* son obligatorios.' At the bottom right are 'Guardar' and 'Eliminar' buttons. The top status bar shows 'Coordinador: a0004', 'Titulación: Ingeniería Informática', 'Asignatura: Redes', 'Año: 09-10', and '24 / 11 / 2009'.

Figura 63: Profesor activado

Desde esta pantalla, el administrador podrá modificar los permisos sobre prácticas del profesor, pulsando nuevamente en el botón “Guardar”. Además podrá eliminar el profesor del sistema pulsando sobre el botón “Eliminar”.

## 2.4.- PROFESORES - BUSCAR/MODIFICAR/BORRAR

El coordinador puede consultar todos los profesores que ha activado y a partir de ahí modificar sus datos y/o eliminarlos del sistema.







Universidad Carlos III de Madrid

Entregador de prácticas

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

Buscar profesores

NIA:   
Nombre:   
Apellidos:

Buscar Mostrar todos

Figura 64: Buscar profesor I

Para consultar los profesores, existen dos opciones:

- Pulsar sobre el botón “Mostrar todos”, de modo que se muestren todos los profesores activados.



Universidad Carlos III de Madrid

Entregador de prácticas

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

Buscar profesores

NIA:   
Nombre:   
Apellidos:

Buscar Mostrar todos

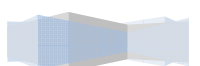
Resultados

Listado de profesores

NIA	Nombre	Apellidos	
100048391	Yankoa	López Cozas	
100048392	Pedro	Ayala Sánchez	
100048393	Ana	López Cozas	

Figura 65: r profesor II

- Escribir el NIA, nombre y/o apellidos que desee buscar y pulsar sobre el botón “Buscar”.





**Entregador de prácticas**

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

**Buscar profesores**

NIA:   
Nombre:   
Apellidos:

**Resultados**

**Listado de profesores**

NIA	Nombre	Apellidos
100048391	Yankoa	López Cozas

Figura 66: r profesor III

En ambas búsquedas, si no existen resultados se mostrará por pantalla el mensaje “No se han encontrados coincidencias”.

Tras la búsqueda, el coordinador puede pulsar sobre el icono de lupa que hay al lado de cada profesor y consultar así sus datos, modificarlos o eliminar al profesor.

## 2.5.- PRÁCTICAS - ALTA

Los coordinadores y profesores con permisos, podrán dar de alta una nueva práctica para una asignatura en un curso académico. Tras pulsar dicha opción aparecerá la siguiente pantalla:



**Entregador de prácticas**

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

**Datos de la práctica**

Nombre\*:  Convocatoria\*: ☐ Ordinaria ☐ Extraordinaria  
Fecha entrega\*:  Hora máxima entrega(Mm:mm)\*:   
Responsable\*:

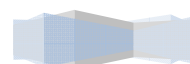
**Autores**

Número de autores\*:   
Comprobar matrícula\*: ☐ Si ☒ No

Los campos marcados con \* son obligatorios.

Figura 67: Alta práctica

Los usuarios deberán indicar los siguientes datos:



- Nombre: Nombre de la práctica.
- Convocatoria: Se debe indicar la convocatoria de la práctica (ordinaria o extraordinaria).
- Fecha de entrega: Se debe indicar la fecha de entrega de la práctica, ya sea marcándola a través del calendario o escribiéndola en formato “dd-mm-aaaa”.
- Hora máxima de entrega: En formato hh:mm.
- Número de autores: Debe ser un número entero comprendido entre 1 y 10.
- Comprobar matrícula: Se debe indicar si es necesario que los grupos de prácticas estén compuestos por alumnos que se encuentren matriculados y por lo tanto en las listas de los profesores.
  - En caso de indicar que debe comprobarse matrícula, es necesario indicar también si se permiten grupos mixtos, es decir, grupos de prácticas con alumnos de diferentes turnos.

**Autores**

Número de autores\*:

Comprobar matrícula\*: ☒ Si ☐ No

Grupos mixtos: ☐ Si ☒ No

Figura 68: Datos autores

Tras introducir los datos, el usuario deberá enviar el formulario pulsando sobre el botón “Guardar”, si existe algún error será informado para que modifique los datos introducidos, en caso contrario aparecerá la siguiente pantalla:

 **Universidad Carlos III de Madrid** **Entregador de prácticas**

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

**Coordinador:**

- Modificar contraseña
- Cambiar de asignatura

**Profesores**

- Activar
- Bucar/Modificar/Borrar

**Prácticas**

- Alta
- Bucar/Modificar/Borrar
- Obtener URL
- Recoger prácticas
- Comprobar firma digital

**Alumnos**

- Importar/Consultar listas
- Consultar grupos

**Cerrar sesión**

 **Los datos se han guardado correctamente.**

**Datos de la práctica**

Nombre\*:  Convocatoria\*: ☒ Ordinaria ☐ Extraordinaria

Fecha entrega\*:  Hora máxima entrega(hh:mm)\*:

Responsable\*:

**Autores**

Número de autores\*:

Comprobar matrícula\*: ☒ Si ☐ No

Grupos mixtos: ☐ Si ☒ No

**Archivos**

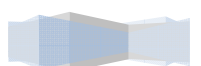
No se han encontrado archivos asociados a la práctica.

[Añadir archivo](#)

Los campos marcados con \* son obligatorios.

[Guardar](#) [Eliminar](#)

Figura 69: Práctica dada de alta



Desde esta pantalla, se podrán modificar todos los datos de la práctica, pulsando nuevamente en el botón “Guardar”. Además podrá eliminar la práctica del sistema pulsando sobre el botón “Eliminar”.

El usuario también podrá indicar que archivos deben entregarse con la práctica y la extensión de los mismos, para ellos debe pulsar sobre el botón “Añadir archivo” de forma que aparezca la siguiente pantalla:

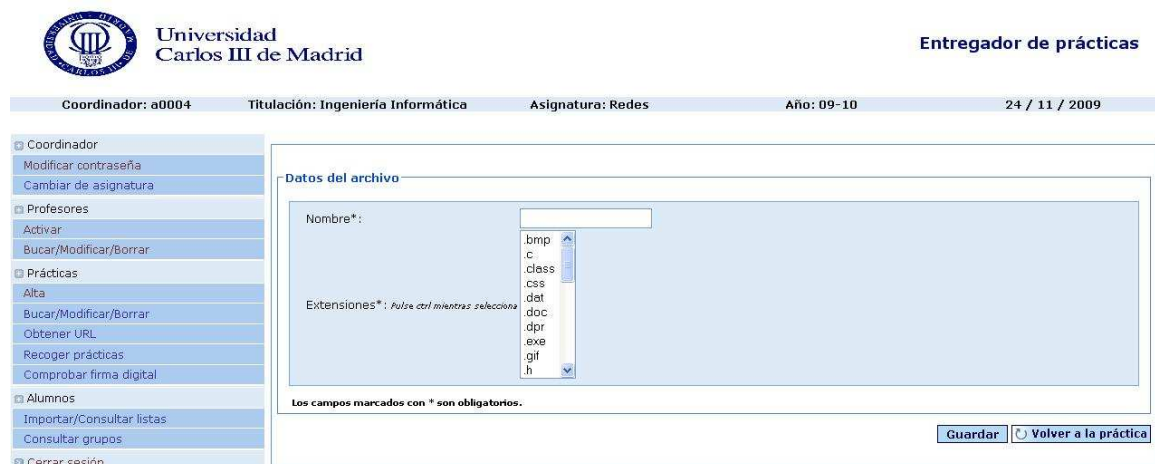


Figura 70: Guardar archivo

En esta pantalla se podrán introducir los datos necesarios para indicar el archivo a entregar:

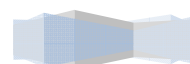
- Nombre: Nombre del archivo se recomienda no poner tildes.
- Extensiones: Se deben seleccionar las posibles extensiones que se admiten para el archivo. Para seleccionar varias, mantenga pulsada la tecla “ctrl” mientras marca las extensiones deseadas.

Tras ello se debe pulsar sobre el botón “Guardar”, a no ser que no se desee almacenar ningún archivo en cuyo caso debe pulsar sobre el botón “Volver a la práctica”.

Si la práctica contiene algún archivo introducido, esta aparecerá en la pantalla junto con sus datos, de forma que si se pulsa sobre la imagen de la lupa se pueda modificar los datos del archivo e incluso borrarlo.



Figura 71: Archivos de una práctica



## 2.6.- PRÁCTICAS – BUSCAR/MODIFICAR/BORRAR

Tanto los coordinadores como los profesores con permisos pueden consultar todas las práctica de la asignatura y a partir de ahí modificar sus datos y/o eliminarlas.



Universidad Carlos III de Madrid

Entregador de prácticas

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

**Buscar prácticas**

Nombre:  Convocatoria: ☐ Ordinaria ☐ Extraordinaria

Fecha entrega:

Responsable:

Figura 72: Buscar práctica I

Para consultar las prácticas, existen dos opciones:

- Pulsar sobre el botón “Mostrar todos”, de modo que se muestren todas las prácticas de la asignatura.



Universidad Carlos III de Madrid

Entregador de prácticas

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

**Buscar prácticas**

Nombre:  Convocatoria: ☐ Ordinaria ☐ Extraordinaria

Fecha entrega:

Responsable:

**Resultados**

**Listado de prácticas**

Nombre	Convocatoria	Fecha tope	Responsable
Practica 1	Ordinaria	27-11-2009	Prueba Prueba Prueba
Prueba	Ordinaria	28-12-2009	Prueba Prueba Prueba

Figura 73: Buscar práctica II

- Escribir el nombre, convocatoria, fecha de entrega y/o el responsable de la práctica/s que desee buscar y pulsar sobre el botón “Buscar”.





Figura 74: Buscar práctica III

En ambas búsquedas, si no existen resultado se mostrará por pantalla el mensaje “No se han encontrados coincidencias”.

Tras la búsqueda, se puede pulsar sobre el icono de lupa que hay al lado de cada práctica y consultar así sus datos, modificarlos o eliminar la práctica.

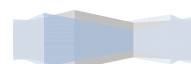
## 2.7.- PRÁCTICAS - OBTENER URL

Tanto los coordinadores como los profesores podrán obtener la URL de una práctica con el fin de incluirla en su página para que los alumnos puedan acceder a la aplicación.

Para ello deben seleccionar en el campo desplegable el nombre de la práctica y la convocatoria de la práctica de la cual desean obtener la URL y pulsar sobre el botón “Buscar”. Tras ello aparecerá debajo la URL de la práctica.



Figura 75: Obtener URL





## 2.8.- PRÁCTICAS – RECOGER PRÁCTICAS

Tanto los coordinadores como los profesores podrán recoger la última versión de todas las prácticas entregadas por los grupos con el fin de visualizarlas y corregirlas

Para ello deben seleccionar en el campo desplegable el nombre de la práctica y la convocatoria de la práctica de la cual desean descargar las entregas y pulsar sobre el botón “Descargar”. Tras ello, si hay prácticas entregadas aparecerá el cuadro de dialogo descargar, para que indique donde desea almacenar el archivo, en caso contrario aparece un mensaje avisando que no existen prácticas entregadas.



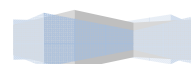
Figura 76: Recoger prácticas entregadas

## 2.9.- PRÁCTICAS – COMPROBAR FIRMA DIGITAL

Coordinadores y profesores, podrán comprobar si una entrega realizada por un grupo de prácticas no ha sido manipulada.

Esta opción debe ser utilizada cuando los alumnos acuden a reclamar, con el fin de comprobar que no han realizado ninguna modificación en la práctica respecto a la entregada.

Para ello, se debe seleccionar el archivo con extensión .zip que contiene la entrega y pulsar sobre el botón “Entregar”, de modo que se visualice el resultado.





Universidad Carlos III de Madrid

Entregador de prácticas

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

☐ Coordinador  
 Modificar contraseña  
 Cambiar de asignatura  
☐ Profesores  
 Activar  
 Bucar/Modificar/Borrar  
☐ Prácticas  
 Alta  
 Bucar/Modificar/Borrar  
 Obtener URL  
 Recoger prácticas  
 Comprobar firma digital  
☐ Alumnos  
 Importar/Consultar listas  
 Consultar grupos  
☐ Cerrar sesión

Entrega realizada por el alumno

Entrega (Archive .zip)\*:  Examinar...

Los campos marcados con \* son obligatorios.

Comprobar

Figura 77: Comprobar firma digital

## 2.10.- ALUMNOS – IMPORTAR/CONSULTAR LISTAS

Los coordinadores y profesores con permisos podrán importar listas de alumnos para cada asignatura. Además profesores y coordinadores, podrán consultar dichas listas.



Universidad Carlos III de Madrid

Entregador de prácticas

Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

☐ Coordinador  
 Modificar contraseña  
 Cambiar de asignatura  
☐ Profesores  
 Activar  
 Bucar/Modificar/Borrar  
☐ Prácticas  
 Alta  
 Bucar/Modificar/Borrar  
 Obtener URL  
 Recoger prácticas  
 Comprobar firma digital  
☐ Alumnos  
 Importar/Consultar listas  
 Consultar grupos  
☐ Cerrar sesión

Listados de alumnos

No se ha encontrado ninguna lista.

Importar lista

Figura 78: Consultar listas

Para importar listas, los profesores con permisos y los coordinadores, deberán pulsar sobre el botón “Importar lista”, apareciendo así la siguiente pantalla:







**Figura 79: Importar lista**

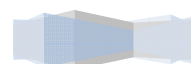
En esta pantalla se podrán introducir los datos necesarios para importar un listado, que son:

- Nombre: Nombre del listado.
- Turno: Se debe seleccionar en el campo desplegable el turno al que corresponde el listado de alumnos.
- Ruta: Se deberá seleccionar el archivo en formato .csv (fichero separado por comas) que contiene el listado de alumnos, pulsando el botón “Examinar” y a través del cuadro de dialogo.

Tras ello se debe pulsar sobre el botón “Guardar”, a no ser que no se desee almacenar ningún listado en cuyo caso debe pulsar sobre el botón “Volver”. Después podrá modificar cualquier dato que desee e incluso borrar la lista importada.

Si la asignatura contiene algún listado importado, en la *figura 78* aparecerá una lista con todos los listados importados, de manera que tras pulsar sobre la imagen de la lupa:

- Los profesores con permisos y coordinadores puedan modificar y/o eliminar los datos. Además si vuelven a pulsar sobre la lupa que aparece en la siguiente pantalla, podrán visualizar los datos del listado de alumnos.



Coordinador: a0004 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

Coordinador

Modificar contraseña

Cambiar de asignatura

Profesores

Activar

Bucar/Modificar/Borrar

Prácticas

Alta

Bucar/Modificar/Borrar

Obtener URL

Recoger prácticas

Comprobar firma digital

Alumnos

Importar/Consultar listas

Consultar grupos

Cerrar sesión

Datos de la lista

Nombre\*: Lista mañana. 

Turno\*: Mañana

Ruta\*:

Los campos marcados con \* son obligatorios.

Figura 80: Visualizar lista I

- Los profesores sin permisos podrán visualizar el listado.

Profesor: a0003 Titulación: Ingeniería Informática Asignatura: Redes Año: 09-10 24 / 11 / 2009

Profesor

Modificar contraseña

Cambiar de asignatura

Prácticas

Obtener URL

Recoger prácticas

Comprobar firma digital

Alumnos

Consultar listas

Consultar grupos

Cerrar sesión

Lista mañana - turno Mañana

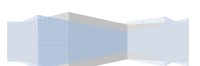
NIA	Nombre	Apellidos
100011111	Juan Ramón	Jiménez Santos
100022222	Ertela	Castillo Bartolome
100033333	Javier	Sanabria Ayala
100044444	Francisco	Olcina Grande

Figura 81: Visualizar lista II

## 2.11.- ALUMNOS – CONSULTAR GRUPOS

Los profesores y coordinadores, podrán consultar los datos y entregas de todos los grupos de prácticas registrados para la asignatura.

Tras pulsar en dicha opción del menú, les aparecerán todos los grupos de prácticas junto con sus componentes.



Coordinador: a0004	Titulación: Ingeniería Informática	Asignatura: Redes	Año: 09-10	24 / 11 / 2009
--------------------	------------------------------------	-------------------	------------	----------------

Coordinador

Modificar contraseña

Cambiar de asignatura

Profesores

Activar

Bucar/Modificar/Borrar

Prácticas

Alta

Bucar/Modificar/Borrar

Obtener URL

Recoger prácticas

Comprobar firma digital

Alumnos

Importar/Consultar listas

Consultar grupos

Cerrar sesión

Consultar grupos de prácticas

g1

NIA	Nombre y apellidos
100022222	Estela Castillo Bartolomé
100033333	Javier Sanabria Ayala

grupo prueba

NIA	Nombre y apellidos
100011111	Juan Ramón Jiménez Santos

Figura 82: Consultar grupos I

Para consultar un grupo mas a fondo, deben pulsar sobre la imagen de la lupa de modo que puedan visualizar las además las entregas realizadas de cada práctica de la asignatura. Siempre que lo deseen podrán también volver atrás pulsando sobre el botón “Volver”.

Coordinador: a0004	Titulación: Ingeniería Informática	Asignatura: Redes	Año: 09-10	24 / 11 / 2009
--------------------	------------------------------------	-------------------	------------	----------------

Coordinador

Modificar contraseña

Cambiar de asignatura

Profesores

Activar

Bucar/Modificar/Borrar

Prácticas

Alta

Bucar/Modificar/Borrar

Obtener URL

Recoger prácticas

Comprobar firma digital

Alumnos

Importar/Consultar listas

Consultar grupos

Cerrar sesión

Integrantes del grupo g1

NIA	Nombre y apellidos
100033333	Javier Sanabria Ayala
100022222	Estela Castillo Bartolomé

Entregas realizadas

Práctica	Fecha	Hora
Prueba	24-11-2009	13:34:54

Volver

Figura 83: Consultar grupos II

Una vez en la pantalla anterior, podrán descargar cualquiera de las prácticas entregadas pulsando sobre la imagen de la bola del mundo que hay al lado de cada práctica entregada. Tras ello, se abrirá el cuadro de dialogo de descarga.

## 2.12.- CERRAR SESIÓN

Tras finalizar de realizar acciones, se recomienda a los coordinadores y profesores que pulsen sobre esta opción para cerrar su sesión y evitar que otros usuarios puedan acceder a ella y modificar sus datos o realizar acciones indebidas.



### 3.- MANUAL DE USUARIO DE ALUMNOS

Mediante la aplicación del entregador de prácticas, cualquier usuario podrá realizar la creación y modificación de grupos de prácticas para una asignatura, además de permitirle la entrega electrónica de prácticas.

Para su utilización, tan solo es necesario un equipo informático conectado a Internet y que posea un navegador Web (se recomienda Internet Explorer 8 o Mozilla Firefox 3 o superior).

Para acceder a dicha aplicación, los profesores de cada asignatura, proporcionan a los estudiantes un enlace a la aplicación para cada una de las prácticas de la asignatura, a través del cual si los alumnos acceden podrán ver la siguiente pantalla:

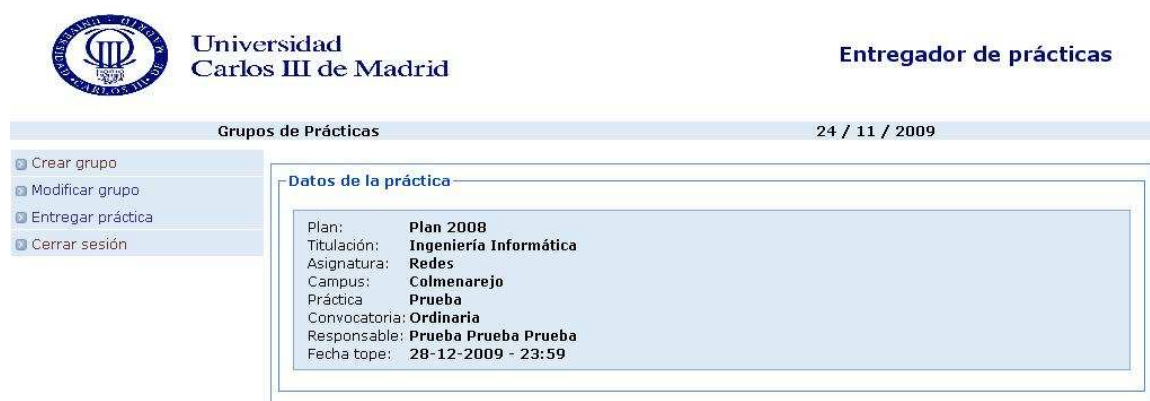


Figura 84: Inicio alumnos

En esta página podrán ver los datos de la práctica, y acceder a cualquiera de las opciones del menú:

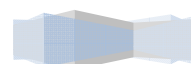
- Crear grupo.
- Modificar grupo.
- Entregar práctica.
- Cerrar sesión.

A continuación, se explicarán cada una de las opciones del menú.

#### 3.1.- CREAR GRUPO

Antes de poder entregar una práctica, los alumnos deben crear un grupo de prácticas para dicha asignatura.

Una vez creado el grupo, los alumnos podrán entregar todas las prácticas correspondientes a una asignatura durante un curso académico completo.



Tras pulsar sobre la opción del menú “Crear grupo”, aparecerá la siguiente pantalla:



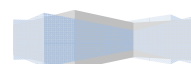
Figura 85: Crear grupo

En esta pantalla, los alumnos deberán indicar los siguientes datos:

- Nombre del grupo: Cadena de caracteres de un tamaño máximo de 20 caracteres que no contenga en ningún caso el guion bajo “\_” ni espacios en blanco. El nombre del grupo no puede existir previamente para dicha asignatura en el mismo curso académico.
- Contraseña: Contraseña de acceso para el grupo de prácticas. Debe contener al menos 7 caracteres y se recomienda la inclusión de números y signos de puntuación para hacerla más segura.
- Repita contraseña: Se debe escribir una segunda vez la contraseña para evitar errores.

Tras completar los datos, se debe enviar el formulario pulsando sobre el botón “Guardar”. En caso de producirse cualquier error, se informará a los alumnos con el fin de que vuelvan a introducir los datos correctamente.

Si los datos han sido introducidos correctamente, aparecerá la siguiente pantalla:



Grupos de Prácticas 24 / 11 / 2009

☐ Crear grupo  
☐ Modificar grupo  
☐ Entregar práctica  
☐ Cerrar sesión

 Los datos se han guardado correctamente.

**Datos de la práctica**

Plan: Plan 2008  
 Titulación: Ingeniería Informática  
 Asignatura: Redes  
 Campus: Colmenarejo  
 Práctica: Prueba  
 Convocatoria: Ordinaria  
 Responsable: Prueba Prueba Prueba  
 Fecha tope: 28-12-2009 - 23:59

**Datos del grupo**

Nombre del grupo\*: grupo prueba  
 Contraseña\*:   
 Repita contraseña\*:

**Alumnos**

No se han encontrado alumnos asociados al grupo de prácticas.

[Añadir alumno](#)

Los campos marcados con \* son obligatorios.

[Guardar](#)

Figura 86: Grupo creado

En esta pantalla, los alumnos podrán modificar la contraseña, cambiando los campos “Contraseña” y “Repita contraseña” y pulsando nuevamente sobre el botón guardar. Además deberán añadir los componentes que forman parte del grupo de prácticas, de lo contrario, no podrán realizar la entrega de la práctica.

Tras pulsar sobre el botón “Añadir alumno” aparecerá la siguiente pantalla:

Grupos de Prácticas 24 / 11 / 2009

☐ Crear grupo  
☐ Modificar grupo  
☐ Entregar práctica  
☐ Cerrar sesión

**Datos de la práctica**

Plan: Plan 2008  
 Titulación: Ingeniería Informática  
 Asignatura: Redes  
 Campus: Colmenarejo  
 Práctica: Prueba  
 Convocatoria: Ordinaria  
 Responsable: Prueba Prueba Prueba  
 Fecha tope: 28-12-2009 - 23:59

**Datos del alumno**

NIA\*:   
 Nombre\*:  Apellidos\*:

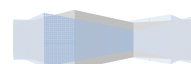
Los campos marcados con \* son obligatorios.

[Guardar](#) [Volver al grupo de prácticas](#)

Figura 87: Añadir alumno al grupo

En esta pantalla, cada miembro del grupo deberá indicar los siguientes datos:

- NIA: Identificador del alumno. Debe comenzar por 100 y tener 9 dígitos.



- Nombre.
- Apellidos.

Tras completar todos los datos, el formulario debe ser enviado pulsando el botón “Guardar”. En caso de producirse cualquier error, se informará a los alumnos con el fin de que vuelvan a introducir los datos correctamente, sino aparecerá la siguiente pantalla.



Universidad Carlos III de Madrid

Entregador de prácticas

Grupos de Prácticas 24 / 11 / 2009

- Crear grupo
- Modificar grupo
- Entregar práctica
- Cerrar sesión

**Datos de la práctica**

Plan: Plan 2008  
Titulación: Ingeniería Informática  
Asignatura: Redes  
Campus: Colmenarejo  
Práctica: Prueba  
Convocatoria: Ordinaria  
Responsable: Prueba Prueba Prueba  
Fecha tope: 28-12-2009 - 23:59

**Datos del grupo**

Nombre del grupo\*: grupo prueba  
Contraseña\*:   
Repita contraseña\*:

**Alumnos**

NIA	Nombre y apellidos	
100011111	Juan Ramón Jiménez Santos	

Añadir alumno

Los campos marcados con \* son obligatorios.

Guardar

Figura 88: Alumno añadido

Si por algún motivo, los alumnos se han confundido y no desean agregar mas componentes a su grupo de prácticas, pueden pulsar sobre el botón “Volver al grupo de prácticas” si realizar ninguna acción.

Tras añadir un componente al grupo, los alumnos podrán pulsar sobre el botón de la lupa que hay al lado de cada componente para visualizar y/o modificar sus datos a excepción del NIA. Además se permite eliminar al componente del grupo de prácticas.





Grupos de Prácticas 24 / 11 / 2009

Crear grupo  
Modificar grupo  
Entregar práctica  
Cerrar sesión

**Datos de la práctica**

Plan: Plan 2008  
Titulación: Ingeniería Informática  
Asignatura: Redes  
Campus: Colmenarejo  
Práctica: Prueba  
Convocatoria: Ordinaria  
Responsable: Prueba Prueba Prueba  
Fecha tope: 28-12-2009 - 23:59

**Datos del alumno**

NIA\*: 100011111  
Nombre\*: Juan Ramón Apellidos\*: Jiménez Santos

Los campos marcados con \* son obligatorios.

Guardar Eliminar Volver al grupo de prácticas

Figura 89: Modificar alumno

### 3.2.- MODIFICAR GRUPO

Tras crear un grupo de prácticas, se ha comprobado que puede modificarse tanto el grupo como sus componentes. Además también existe esta opción en el menú para poder acceder posteriormente a realizar cambios.

Cuando los alumnos pulsán sobre esta opción pueden darse dos situaciones:

- Que el grupo este identificado en el sistema, de modo que aparezca la pantalla con los datos del grupo y se permita la modificación de la contraseña o de los componentes del grupos.
- O por el contrario, que el grupo no esté identificado, y deba identificarse para poder realizar modificaciones. En este caso, deberán introducir el nombre del grupo y la contraseña, y si ambos datos son correctos, se mostrara la pantalla de modificación.

Grupos de Prácticas 10 / 11 / 2009

Crear grupo  
Modificar grupo  
Entregar práctica  
Cerrar sesión

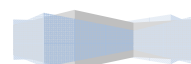
**Acceso grupos**

Nombre del grupo\*:  
Contraseña\*:

Los campos marcados con \* son obligatorios.

Entrar

Figura 90: Validación grupos





### 3.3.- ENTREGAR PRÁCTICA

Cuando los alumnos pulsán sobre esta opción pueden darse dos situaciones:

- Que el grupo este identificado en el sistema, de modo que aparezca directamente la pantalla de entrega.
- O por el contrario, que el grupo no esté identificado, y deba identificarse para poder realizar la entrega. En este caso, deberán introducir el nombre del grupo y la contraseña, y si ambos datos son correctos, se mostrara la pantalla de entrega.

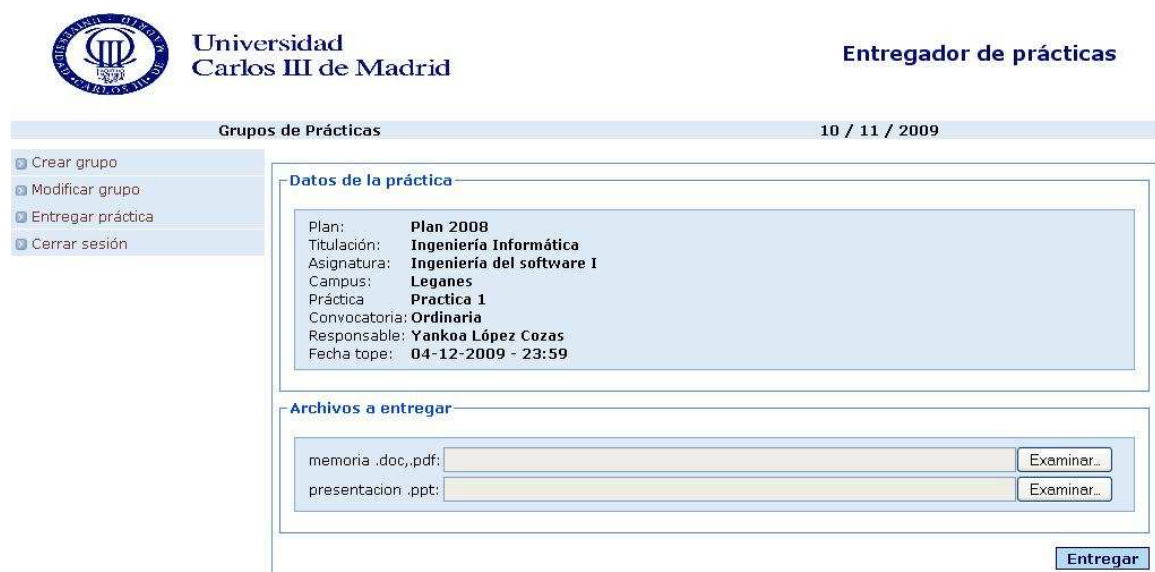
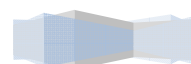


Figura 91: Entregar práctica I

Para poder realizar la entrega, los alumnos deberán seleccionar los archivos a entregar, pulsando sobre el botón “Examinar” de cada archivo. Tras pulsar sobre el botón les aparecerá un cuadro de dialogo para explorar su sistema de ficheros y donde deberán seleccionar el archivo a entregar.

El archivo debe tener el mismo nombre y extensión que el indicado. Si aparecen dos o más extensiones, el archivo a entregar puede ser de cualquiera de dichas extensiones.

Una vez seleccionados todos los archivos a entregar, los alumnos deben pulsar sobre el botón “Entregar”, de modo que si no se produce ningún error aparecerá la siguiente pantalla:



Grupos de Prácticas


10 / 11 / 2009

Crear grupo

Modificar grupo

Entregar práctica

Cerrar sesión

 Los archivos se han entregado correctamente.

Datos de la práctica

Plan: Plan 2008

Titulación: Ingeniería Informática

Asignatura: Ingeniería del software I

Campus: Leganes

Práctica: Practica 1

Convocatoria: Ordinaria

Responsable: Yankoa López Cozas

Fecha tope: 04-12-2009 - 23:59

Para descargar la entrega y conseguir el fichero firmado pulse en el botón descargar.

Tenga en cuenta que si no realiza la descarga, no podrá realizar ninguna reclamación posterior.

Descargar

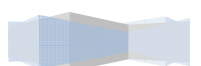
Figura 92: Entregar práctica II

En esta pantalla, los alumnos podrán ver como la práctica ha sido entregada correctamente y tendrán opción de descargarse el resguardo de la práctica pulsando sobre el botón “Descargar”.

El resguardo es un fichero con extensión .zip de nombre <fechaEntrega>\_<nombreGrupo>, que contiene la entrega realizada y la firma de dicha entrega, con el los alumnos podrán realizar cualquier reclamación que consideren necesaria, por lo tanto si no se lo descargan perderán el derecho a reclamar.

### 3.4.- CERRAR SESIÓN

Tras finalizar de realizar acciones, se recomienda a los alumnos que pulsen sobre esta opción para cerrar su sesión y evitar que otros alumnos puedan acceder a ella y modificar sus datos o realizar entregas.



## ANEXO II.- IMPLANTACIÓN EN ENTORNOS WINDOWS

En este apartado se describirá el plan de implantación del sistema en entornos con plataforma Windows, describiéndose los pasos necesarios para la correcta instalación de todos los programas necesarios para el funcionamiento de la aplicación.

La aplicación tiene dos modos de instalación:

- **Modo estándar:** Simplemente se despliega la aplicación en el servidor, sin modificar ninguna parte del código y a través de un fichero .war.
- **Modo personalizado:** Se configuran los parámetros necesarios y se genera el .war que después deberá ser desplegado en un servidor.

### 1.- FORMACIÓN NECESARIA PARA LA IMPLANTACIÓN

Véase apartado 7.1.- *Formación necesaria para la implantación*, sustituyendo Linux por Windows.

### 2.- INFRAESTRUCTURA

Véase apartado 7.2.- *Infraestructura*.

### 3.- ASPECTOS HARDWARE

Véase apartado 7.3.- *Aspectos Hardware*.

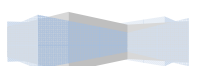
### 4.- ASPECTOS SOFTWARE

La aplicación ha sido implementada y probada en equipos que contaban con Microsoft Windows XP, por lo tanto, se recomienda su instalación en dicho entorno.

Tanto para el modo estándar como para el personalizado, es necesario instalar las siguientes aplicaciones.

- Java Runtime Enviroment (JRE).
- MySQL.
- Apache Tomcat.

Además para el modo personalizado, es conveniente la instalación del entorno de desarrollo Eclipse.



#### 4.1.- JAVA RUNTIME ENVIROMENT 6 UPDATE 17 (JRE)

JRE es un conjunto de utilidades que permiten la ejecución de programas Java.

Para llevar a cabo su instalación, podemos ejecutar el archivo contenido en el CD del proyecto en la ruta “*Windows/Programas/jre-6u17-windows-i586*” que incluye la versión 6 update 17, si lo deseamos podemos descargar dicho archivo de la siguiente dirección:

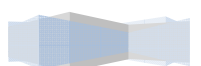
*<http://java.sun.com/javase/downloads/index.jsp>*

Durante la instalación deberemos seguir los siguientes pasos:

- 1) Pulsaremos sobre el botón “Instalar”. Se recomienda dejar la ruta por defecto.



Figura 93: Instalación JRE I



- 2) A continuación comenzará el proceso de instalación.

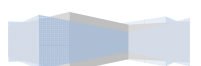


Figura 94: Instalación JRE II

- 3) Finalmente cuando se haya instalado pulsaremos sobre el botón cerrar.



Figura 95: Instalación JRE III



## 4.2.- MySQL 5.1

MySQL es el gestor de bases de datos utilizado en la aplicación, para su instalación, podemos ejecutar el archivo contenido en el CD del proyecto en la ruta “*Windows/Programas/mysql-essential-5.1.37-win32*” o si lo deseamos podemos descargar dicho archivo de la siguiente dirección:

*<http://dev.mysql.com/downloads/mysql/5.1.html>*

Durante la instalación deberemos seguir los siguientes pasos:

- 1) Primeramente aparecerá una pantalla de bienvenida que nos advertirá que el programa está protegido con Copyright. Pulsamos “Next”.

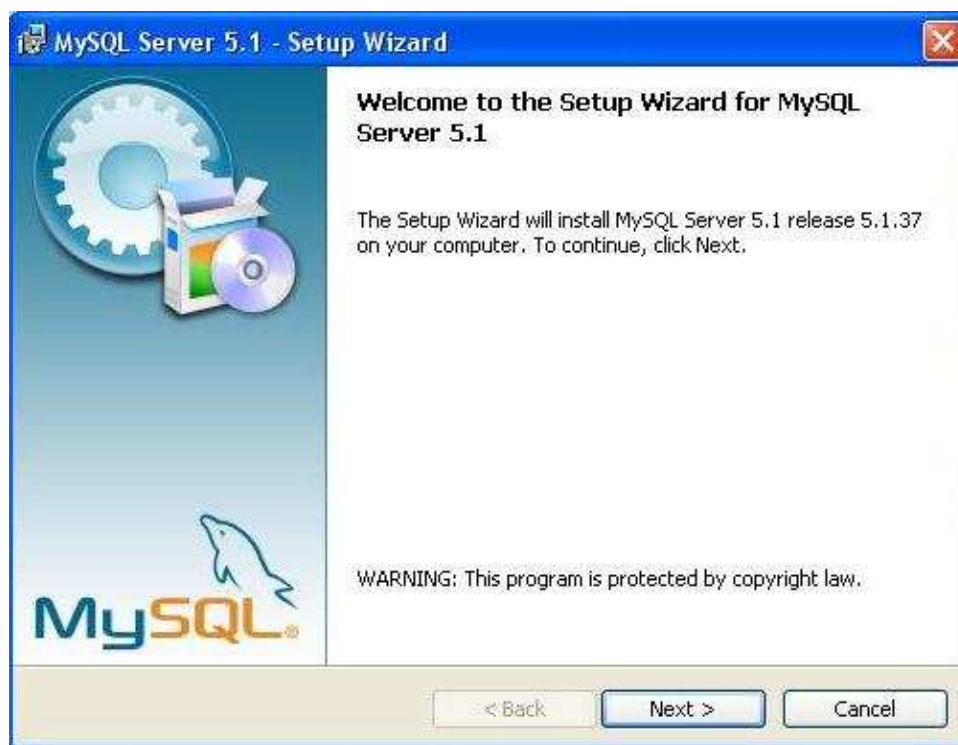
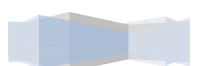


Figura 96: Instalación MySQL I





- 2) La siguiente pantalla nos pedirá que seleccionemos el tipo de instalación que deseamos. En este caso seleccionamos la instalación típica y pulsamos “Next”.

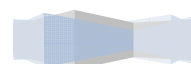


Figura 97: Instalación MySQL II

- 3) En la siguiente pantalla nos indicará el tipo de instalación y la ruta donde va a instalarse. Pulsamos “Install” para que comience el asistente de instalación.



Figura 98: Instalación MySQL III



- 4) A continuación comenzará el proceso de instalación.

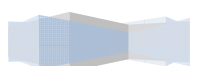


Figura 99: Instalación MySQL IV

- 5) En las siguientes pantallas, se muestra una breve explicación del programa. Pulsaremos en todas “Next”.



Figura 100: Instalación MySQL V





- 6) Una vez finalizada la instalación, deberemos configurar MySQL. Marcaremos la opción “*Configure the MySQL Server now*”, para que se abra el asistente de configuración, después pulsaremos “Finish”.

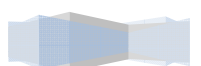


Figura 101: Instalación MySQL VI

- 7) Aparecerá una pantalla que te da la bienvenida al asistente de configuración. Pulsamos “Next”.



Figura 102: Instalación MySQL VII



- 8) Ahora nos preguntará que tipo de configuración deseamos. Seleccionaremos la configuración estándar y pulsaremos “Next”.

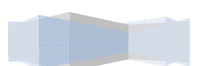


Figura 103: Instalación MySQL VIII

- 9) La siguiente pantalla configura el servicio de MySQL. Dejamos todos como está y pulsamos “Next”.



Figura 104: Instalación MySQL IX



- 10) Ahora deberemos crear el administrador de la base de datos. Seleccionamos “Modify Security Settings”, introducimos la contraseña que deseemos y la confirmamos volviéndola a escribir. En caso de ejecutar el programa en modo estándar, la contraseña debe ser “yankoa”. Después pulsamos “Next”.



Figura 105: Instalación MySQL X

- 11) Una vez hemos terminado de configurar MySQL, aparecerá una pantalla donde se nos pedirá que ejecutemos la configuración para ver si todo está correcto. Pulsamos “Execute”. Una vez se haya ejecutado todo. Si no ha ocurrido ningún problema, aparecerá la siguiente pantalla, pulsaremos “Finish” y todo el proceso habrá terminado.  
En caso de producirse algún error, puede ser debido al firewall, o algún antivirus que bloquee el servicio.

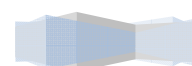




Figura 106: Instalación MySQL XI

### 4.3.- APACHE TOMCAT 5.5

Apache Tomcat es el servidor de aplicaciones utilizado para poner en marcha la aplicación, para su instalación, podemos ejecutar el archivo contenido en el CD del proyecto en la ruta “*Windows/Programas/apache-tomcat-5.5.28*” o si lo deseamos podemos descargar la distribución Core de la siguiente dirección:

*<http://tomcat.apache.org/download-55.cgi>*

Durante la instalación deberemos seguir los siguientes pasos:

- 1) En la página de bienvenida pulsaremos “Next”.

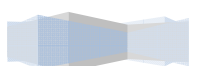






Figura 107: Instalación Apache Tomcat I

- 2) Aceptaremos la licencia del programa pulsando “I Agree”.

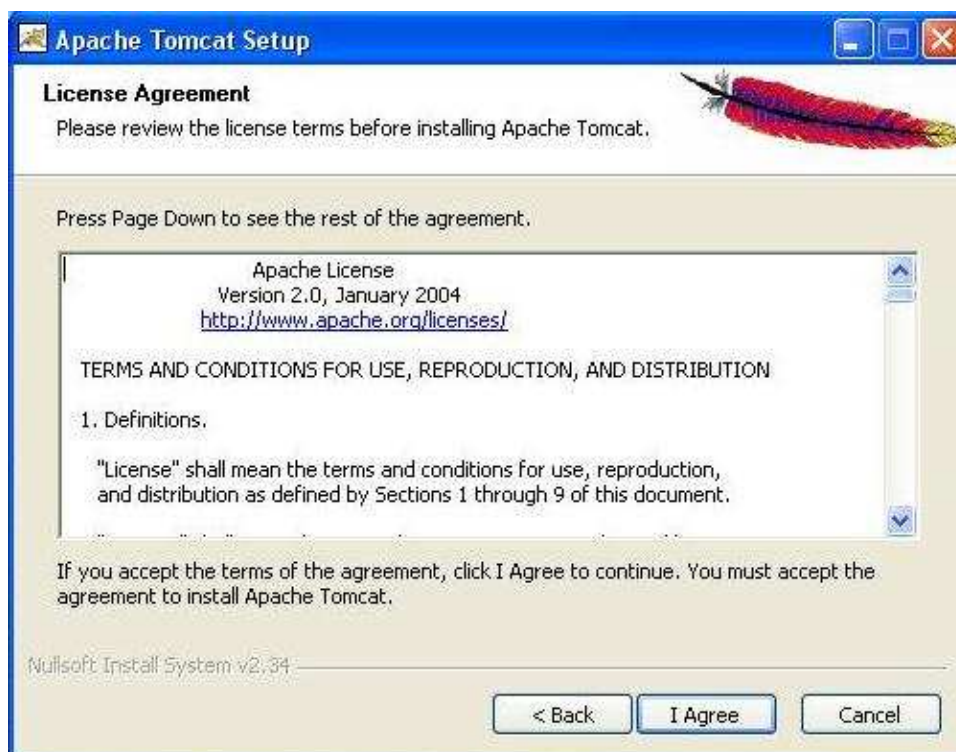
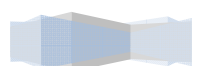


Figura 108: Instalación Apache Tomcat II



- 3) Después nos pedirá que seleccionemos el tipo de instalación y los componentes que deseamos instalar. Seleccionaremos la instalación propia y todos los componentes, excepto los ejemplos. Después pulsaremos “Next”.

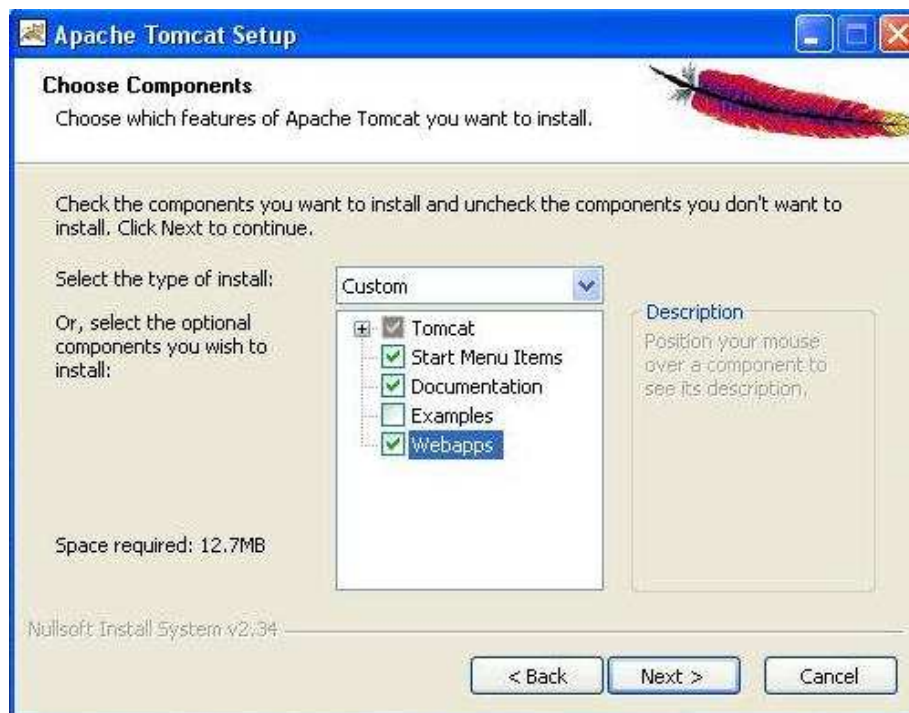


Figura 109: Instalación Apache Tomcat III

- 4) Después elegiremos la ruta donde deseamos instalar el programa y pulsaremos “Next”. Se recomienda dejar la ruta por defecto.

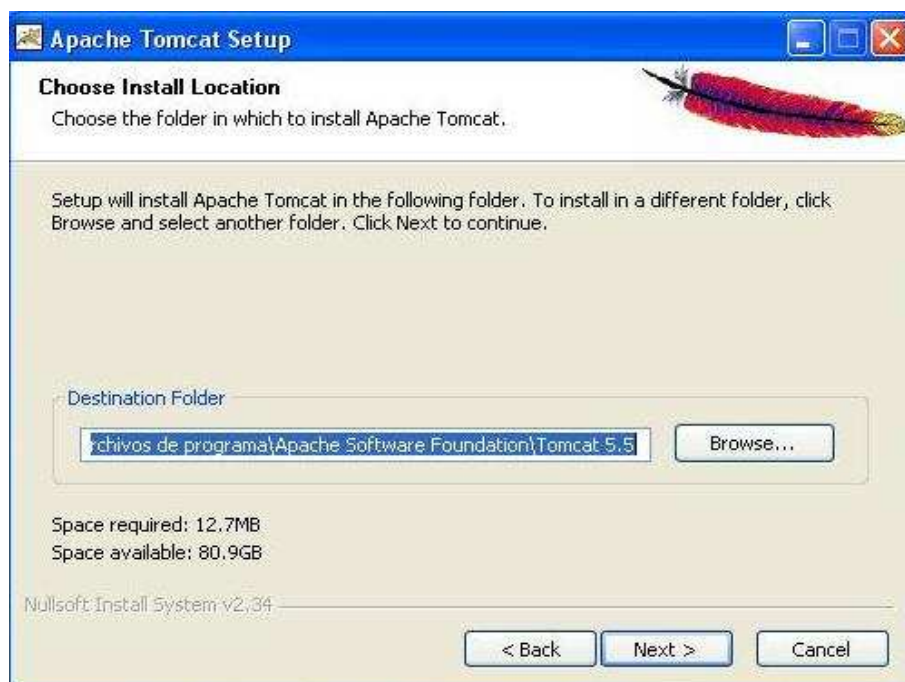
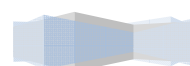


Figura 110: Instalación Apache Tomcat IV



- 5) Ahora deberemos introducir los parámetros básicos de configuración de Tomcat. Primero introduciremos el puerto de Internet por el que correrá Tomcat, el 8080. Después pondremos el nombre de usuario de Tomcat, por ejemplo, admin. La contraseña no es necesaria. Finalmente pulsaremos “Next”.

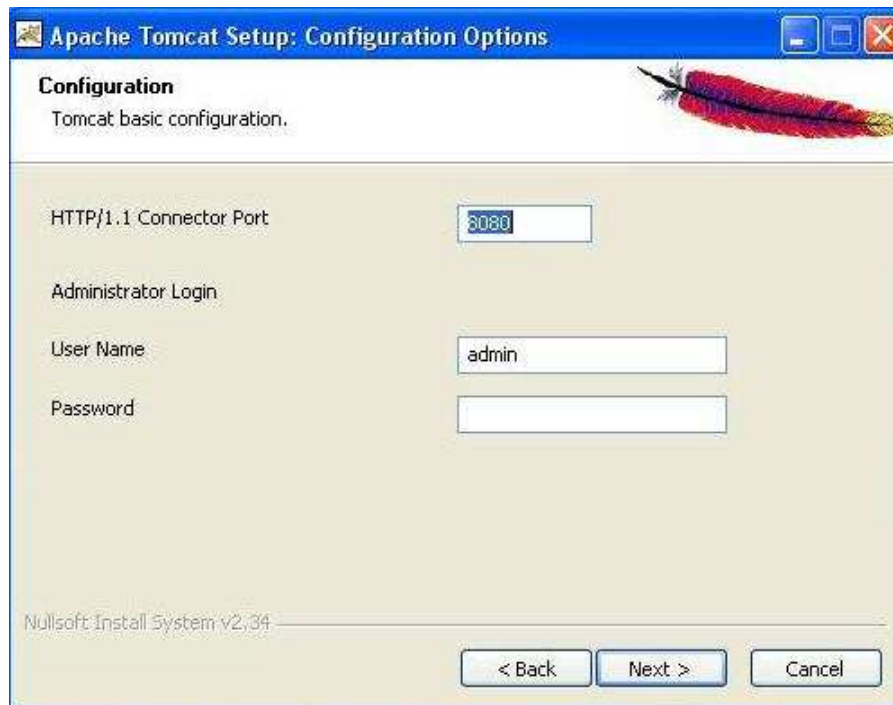


Figura 111: Instalación Apache Tomcat V

- 6) Ahora nos pedirá que introduzcamos la ruta en la que se encuentra en nuestro equipo la Máquina Virtual de Java (JRE). Deberemos introducir la ruta en la que la instalamos y pulsar “Install”.

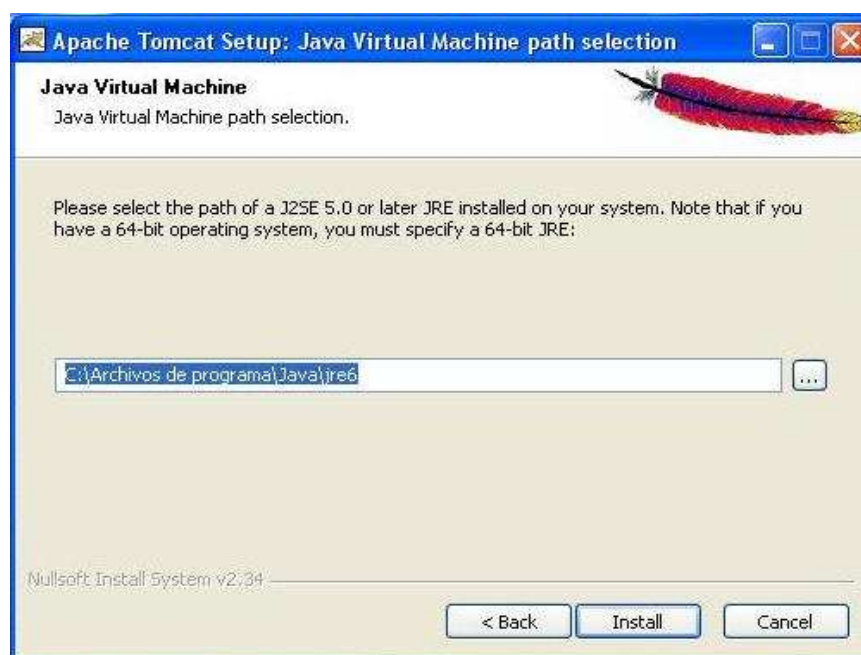
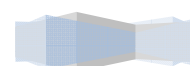


Figura 112: Instalación Apache Tomcat VI



- 7) En este momento comenzará el proceso de instalación.

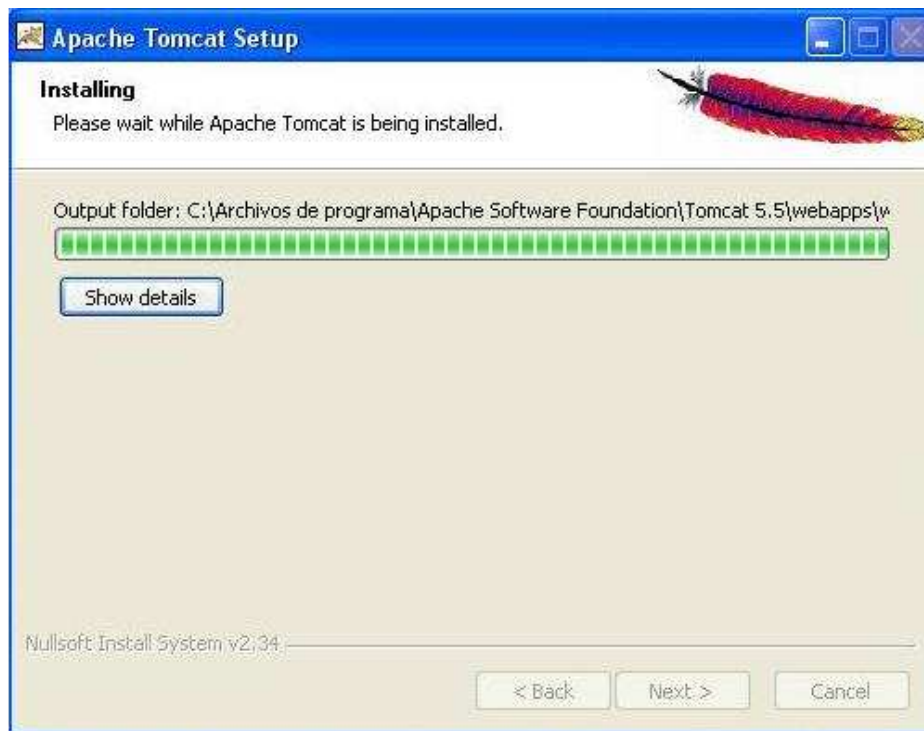


Figura 113: Instalación Apache Tomcat VII

- 8) Finalmente se mostrara como el proceso se ha completado. Desmarcaremos todas las opciones y pulsaremos sobre “Finish”.

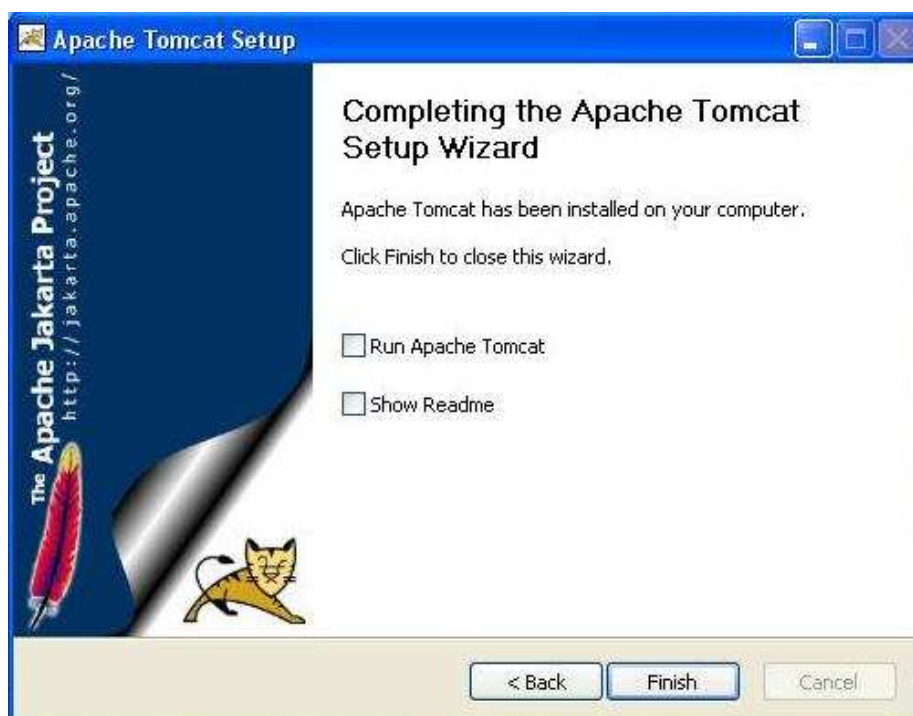
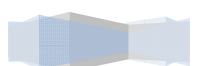


Figura 114: Instalación Apache Tomcat VIII





## 5.- EJECUCIÓN EN MODO ESTÁNDAR

Tras la instalación de los programas necesarios, es posible ejecutar el programa en modo estándar, es decir, mediante la configuración por defecto.

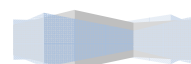
Para ello, deben seguirse los siguientes pasos:

- 2) Importar la base de datos en MySQL. Para ello debemos abrir la consola de comandos de MySQL, que se encuentra en el menú *“Inicio/Todos los programas/MySQL/MySQL Server 5.1/MySQL Command Line Client”* y escribir lo siguiente:
  - a. Contraseña, que en este caso debería ser “yankoa”.
  - b. Importar la BD, mediante el siguiente comando *“source ruta/archivo.sql”* donde ruta es la ruta donde se encuentra la BD y archivo el nombre del archivo. Este archivo se encuentra en el CD del proyecto en la siguiente ruta *“BD/Entregador.sql”*.
- 3) Se debe colocar el conector de la BD dentro de las librerías de Tomcat, para que de esta forma se pueda realizar la conexión con la BD cargada anteriormente. Debemos seguir los siguiente pasos:
  - a. Descomprimir el archivo *mysql-connector-java-5.1.8.zip* contenido en el CD del proyecto en la carpeta *“BD”*.
  - b. Copiar el archivo *mysql-connector-java-5.1.8-bin.jar* en las librerías de Tomcat, en la siguiente ruta *“C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\common\lib”*.
- 4) Generar el almacén de claves o utilizar el existente.
  - a. Si desea generar un almacén de claves, debe abrir una consola de comandos y colocarse en la ruta donde se encuentra instalada su máquina virtual. Tras ello, debe teclear el siguiente comando:

*keytool -genkey -alias tomcat -keyalg RSA*

Después debe rellenar todos los datos necesarios, y utilizar como contraseña la palabra “changeit”. Tras ello, se generará un fichero de claves con el nombre “keystore” en la siguiente ruta *“C:\Documents and Settings\MiNombreDeUsuario”*.

- b. Tanto si ha generado el fichero de claves, como si desea hacer uso del que se encuentra en el CD del proyecto en la siguiente ruta *“Almacen claves/keystore”*, debe colocar dicho fichero en *“C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5”*, para conseguir que el programa funcione correctamente



- 5) Configurar el fichero del servidor Tomcat (server.xml) que se encuentra en la ruta “C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\conf”, descomentando la parte donde se crea un servidor seguro, es decir, SSL.

- a. La parte a descomentar es la siguiente:

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->
<Connector port="8443" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25"
    maxSpareThreads="75" enableLookups="false"
    disableUploadTimeout="true"
    acceptCount="100" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"/>
```

- b. Al final de ella, debemos añadir el lugar donde se encuentra el fichero de claves y la contraseña. Mediante la siguiente línea:

```
keystoreFile="C:\Archivos de programa\Apache Software
Foundation\Tomcat 5.5\" keystorePass="changeit"/>
```

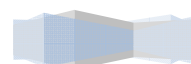
- 6) Colocar el archivo con extensión .war que se encuentra en el CD del proyecto “Windows/War/Entregador.war” dentro de la carpeta webapps de Tomcat, que debe encontrarse en la siguiente ruta “C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5”. De esta manera, el servidor de aplicaciones, logrará desplegarla para su ejecución.
- 7) Una vez hechos todos los pasos, debemos comenzar el servicio de Tomcat pulsando start, y ya podremos probar la aplicación. Se destaca que todos los archivos del servidor, se almacenarán en el siguiente directorio “C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\Entregador”, en caso de no desear que se almacenen en esa ruta, deberá seguir las instrucciones para ejecutar en modo personalizado.

## 6.- EJECUCIÓN EN MODO PERSONALIZADO

Tras la instalación de los programas, es posible ejecutar la aplicación de un modo personalizado, esto quiere decir, que deberá crear el fichero .war una vez se hayan configurado todas las rutas por defecto.

Para modificar las rutas por defecto, basta con abrir los ficheros donde se encuentran los valores a modificar, una opción es hacerlo manualmente desde cualquier editor de texto, aunque es mas sencillo desde el entorno de desarrollo integrado Eclipse.

En primer lugar, puede que al generar la BD, haya indicado una contraseña diferente a la indicada anteriormente, por lo tanto, deberá modificar la contraseña del fichero de configuración de Hibernate para que la aplicación se ejecute correctamente.



Para ello debe modificar el fichero hibernate.cfg.xml que se encuentra en el CD del proyecto en la ruta “Codigo\Entregador\src” y modificar la siguiente línea cambiando la contraseña:

```
<property name="connection.password">yankoa</property>
```

Tras ello, puede llevar a cabo los pasos 1 y 2 del apartado 5.- *Ejecución modo estándar*.

Por otro lado, también puede desear, generar un nuevo almacén de claves siguiendo el paso 3 del apartado 5.- *Ejecución modo estándar* y que desee colocarlo en una ruta diferente a la mencionada en ese apartado. Si es así, después de generar el fichero y colocarlo en la ruta deseada, debe abrir el fichero con el nombre Constantes.java que se encuentra en el CD del proyecto en “Codigo\Entregador\src\utilidades” y modificar la siguiente constante, indicando la ruta donde usted haya incluido el almacén de claves:

```
public static final String KEYSTORE = "C:\\Archivos de programa\\Apache  
Software Foundation\\Tomcat 5.5\\.keystore";
```

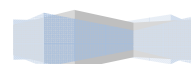
Después debe realizar el paso 4 del apartado 5.- *Ejecución modo estándar* añadiendo en las líneas “keystoreFile” la ruta que ha indicado.

Finalmente, puede ser que desee modificar la ruta donde se almacenan los ficheros generados en el servidor, si es así, debe modificar el archivo Constantes.java que se encuentra en el CD del proyecto en “Codigo\Entregador\src\utilidades” y modificar la siguiente constante, indicando la ruta donde usted desee almacenar todos los datos:

```
public static final String PROYECTO = "Entregador\\";
```

Una vez configurados todos los elementos, se debe proceder a generar el fichero .war para poder desplegarlo en el servidor, para ello tenemos dos opciones.

- Si nos encontramos dentro de Eclipse, debemos pulsar con el botón derecho en el proyecto sobre la opción exportar a war. Aquí elegimos el nombre del war, que se recomienda sea Entregador, la ruta donde deseamos almacenarlo, además de indicar el servidor Tomcat 5.5.



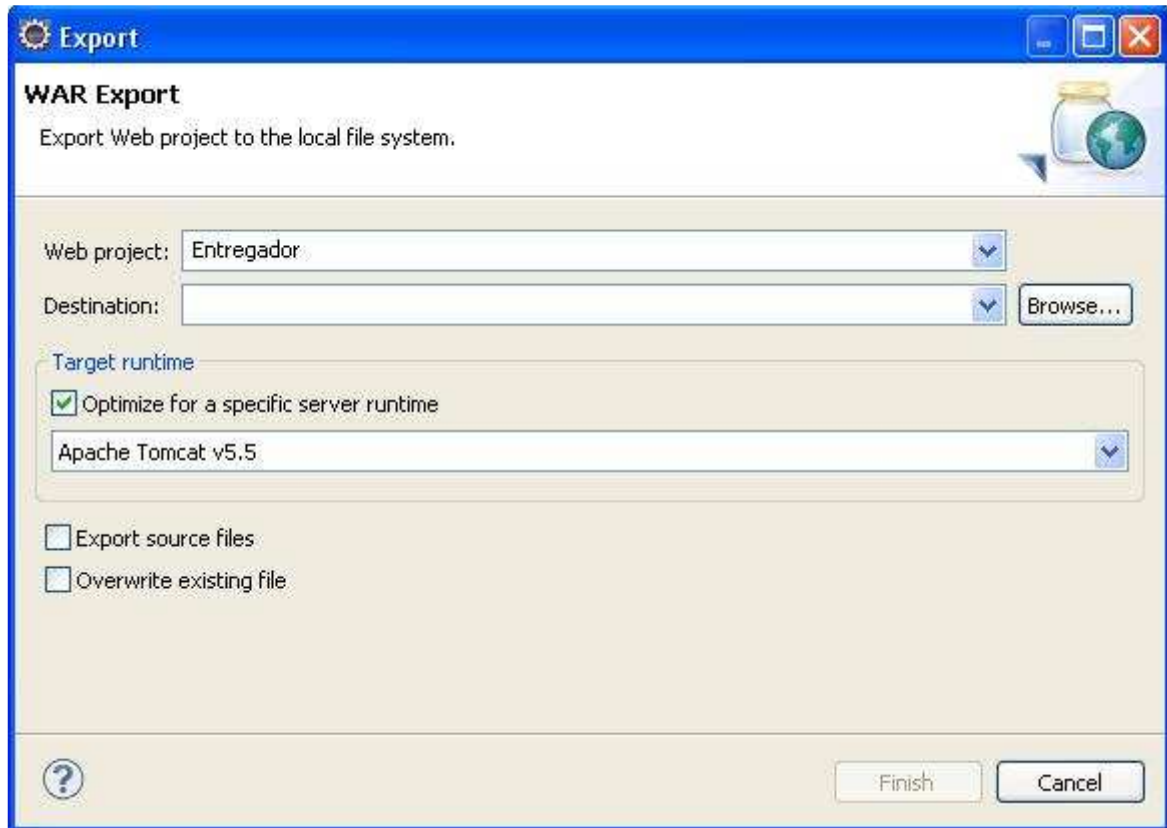


Figura 115: Creación War desde Eclipse

- Si no deseamos usar Eclipse, se puede generar el war desde la línea de comando, situándonos en el directorio del proyecto, debemos teclear lo siguiente:

```
jar cvf Entregador.war *
```

Cuando ya hemos generado el war, debemos seguir los pasos 5 y 6 del apartado 5.- *Ejecución modo estándar*.

## 7.- POSIBLES ERRORES

Véase apartado 7.7.- *Posibles Errores*.



## ANEXO III.- EL MARCO DE TRABAJO STRUTS

A continuación se describirán brevemente los tres componentes que forman parte del MVC, dentro de Struts, además de algunos otros elementos que completan Struts mejorando aspectos básicos de las aplicaciones Web en general.

### 1.- CONTROLADOR DE STRUTS

El controlador dentro de una aplicación MVC se encarga de recibir datos de un cliente, invocar una operación de negocio y coordinar la vista a devolver al cliente, por supuesto hay mas acciones, pero estas son las principales.

El marco de trabajo Struts utiliza un servlet para procesar las peticiones entrantes, además de muchos otros componentes que vamos a examinar detenidamente.

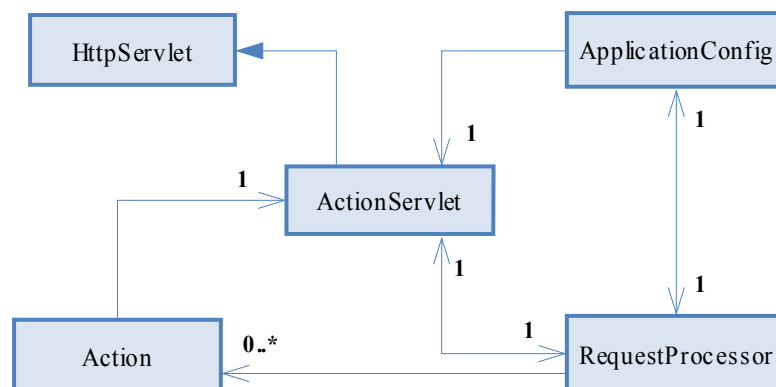


Figura 116: Componentes del controlador de Struts

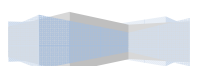
#### 1.1.- LA CLASE ACTIONSERVLET

La clase *org.apache.struts.action.ActionServlet* actúa como un interceptor en las aplicaciones Struts y es responsable de empaquetar y enrutar el tráfico HTTP hasta el manejador adecuado.

Cuando una petición del cliente pasa por ActionServlet, se invoca al método *process()* para gestionarla.

#### 1.2.- LA CLASE REQUEST PROCESSOR

Tras gestionar la petición, el siguiente paso es llamar al método *process()* de la clase *org.apache.struts.action.RequestProcessor*.



Esta clase permite que los desarrolladores personalicen el comportamiento de gestión de petición para una aplicación.

### 1.3.- LA CLASE ACTION

---

La clase *org.apache.struts.action.Action* es el enlace entre una petición del cliente y una operación de negocio. Además también puede realizar funciones de autorización, conexión, validación...

Las clases Action llevan a cabo una sola operación de negocio, es decir, que puede realizar todas las tareas que pertenecen a una única funcionalidad.

Esta clase tiene muchos métodos, pero el más importante es el método *execute()*, que lo invoca la clase RequestProcessor cuando recibe una petición del cliente, de forma que el controlador crea una instancia de la clase Action si no existe ya alguna.

### 1.4.- LA CLASE ACTIONFORWARD

---

La clase *org.apache.struts.action.ActionForward* representa una abstracción lógica de un recurso Web (normalmente una JSP).

El objeto ActionForward contiene detalles que indican donde debe el ActionServlet enviar el control para proporcionar una respuesta adecuada.

## 2.- MODELO DE STRUTS

---

El modelo representa los datos de negocio de una aplicación, por ello se incluyen en el las entidades de negocio y las reglas que gobiernan el acceso y modificación de los datos. Es muy importante que todo ello se mantenga en una única ubicación con el fin de mantener la integridad de los datos, reducir la redundancia y aumentar la reutilidad.

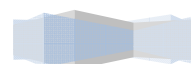
### 2.1.- LOS OBJETOS DE NEGOCIO

---

Los objetos de negocio que forman parte del modelo, son una abstracción software de una entidad del mundo real. Estos objetos constan de un estado y un comportamiento.

Para que una clase Java sea considerada como un objeto de negocio debe cumplir las siguientes condiciones:

- Tener estado y comportamiento.
- Representar una entidad del mundo real.
- Ser reutilizable.



## 2.2.- LA PERSISTENCIA

Se entiende por persistencia en la programación orientada a objetos, la capacidad que tienen los objetos de conservar su estado e identidad entre distintas ejecuciones del programa que los creó o de otros programas que accedan a ellos, es decir, que los datos que se incorporan a una aplicación existirán más allá de la vida de esta.

Cuando los objetos se crean en memoria para una aplicación, no pueden estar allí para siempre, con el tiempo tienen que ser eliminados o persistir en un almacén de datos. Una vez que los objetos han persistido, se pueden recuperar y utilizar para reconstruir los objetos de negocio en un futuro.

La persistencia permite al programador almacenar, transferir y recuperar el estado de los objetos. Para ello existen varias técnicas:

- La **serialización** consiste en un proceso de codificación de un objeto en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato legible como XML. La serie de bytes o el formato pueden ser usados para re-crear un objeto que es idéntico en su estado interno al objeto original. La serialización es un mecanismo ampliamente usado para transportar objetos a través de una red, para hacer persistente un objeto en un archivo o base de datos, o para distribuir objetos idénticos a varias aplicaciones o localizaciones.
- Los **motores de persistencia** traducen los registros a objetos y viceversa. Cuando el programa quiere grabar un objeto llama al motor de persistencia, que traduce el objeto a registros y llama a la base de datos para que guarde estos registros. De la misma manera, cuando el programa quiere recuperar un objeto, la base de datos recupera los registros correspondientes, los cuales son traducidos en formato de objeto por el motor de persistencia.

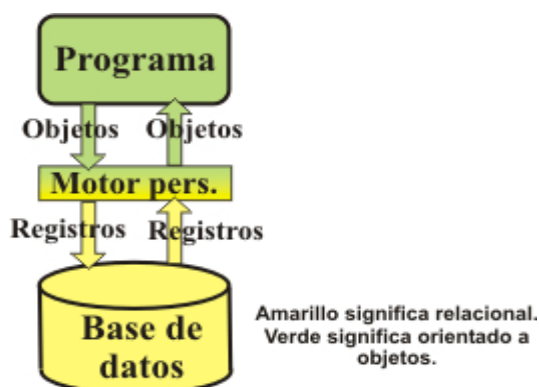
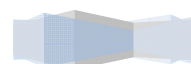


Figura 117: Motor de persistencia

- Las **bases de datos** orientadas a objetos, son aquellas en las que la información se representa mediante objetos como los presentes en la programación orientada a objetos. Cuando se integra las características de una base de datos



con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos (ODBMS). Un ODBMS hace que los objetos de la base de datos aparezcan como objetos de un lenguaje de programación en uno o más lenguajes de programación a los que dé soporte.

Para crear una aplicación Web utilizando Struts, es necesario mapear los objetos de negocio a la base de datos, así según nuestras necesidades podremos usar diversos motores de persistencia, crear uno propio...

### 2.3.- LOS PATRONES BUSSINESS DELEGATE Y DATA ACCESS OBJECT

Al utilizar un motor de persistencia, es necesario desacoplar la persistencia todo lo que sea posible de la aplicación, para ello es necesario crear una interfaz de servicio que pueda utilizar la clase Action para que esta clase no interactúe directamente con el marco de persistencia.

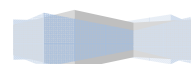
El patrón **Data Access Object** (DAO) se utiliza para desacoplar la lógica de negocio del acceso a datos, además de permitir que la implementación de persistencia sea fácilmente reemplazada por otra sin afectar negativamente a los objetos de negocio.

El patrón **Business Delegate** se encarga de crear y gestionar los objetos de lógica de negocio además de proporcionar un interfaz con los métodos de negocio que se pueden utilizar.

## 3.- VISTA DE STRUTS

Struts utiliza los componentes de vista para mostrar contenido dinámico al cliente. Los principales componentes de vista que se emplean en Struts son:

- Documentos HTML.
- Librerías de etiquetas personalizadas JSP.
- JavaScript y hojas de estilo.
- Archivos multimedia.
- Paquetes de recurso de mensaje, los cuales proporcionan un medio para soportar la localización y ayudan a reducir el tiempo de mantenimiento y la redundancia en toda la aplicación.
- Clases ActionForm.





### 3.1.- USO DE JAVA BEANS

---

La arquitectura de JavaBeans permite a los desarrolladores crear componentes reutilizables que se puedan usar en cualquier plataforma que soporte una máquina virtual de Java. El modelo JavaBeans soporta propiedades, eventos, métodos y persistencia.

Dentro de Struts solo se utilizan algunas de las posibilidades de JavaBeans. Para ello se usa el patrón Objeto de Transferencia de Datos (DTO), el cual agrega varios atributos de una o más entidades de negocio y sitúa esta información en una instancia JavaBean mediante la cual se puede recuperar toda la información de un objeto sin sufrir ningún impacto negativo en el rendimiento de la aplicación.

Como resumen, podemos decir que los DTO son copias locales de los objetos de negocio que son usadas por la vista para entregar los datos de modelo que se muestran con la información estática en la página.

### 3.2.- LA CLASE ACTIONFORM

---

Los objetos de la clase ActionForm se utilizan para capturar los datos de entrada de un formulario HTML y transferirlos a la clase Action. De esta forma el marco de trabajo recopila automáticamente la entrada de datos de la petición y pasa estos datos a una Action utilizando un form bean, que luego pasa a la capa de negocio.

La clase ActionForm actúa como un buffer para albergar el estado de los datos que nos ha facilitado el usuario y además actúa como cortafuegos ya que ayuda a mantener la entrada de datos no válida o sospechosa fuera de su nivel de negocio hasta que se hayan analizado las reglas de validación que se pueden crear dentro de esta clase.

Para toda página HTML en la que se utilicen formularios se deberá usar una ActionForm, así la misma ActionForm se puede utilizar para múltiples páginas si fuera necesario, siempre y cuando coincidan los campos HTML y las propiedades ActionForm.

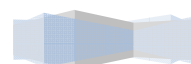
Los ActionForm pueden tener dos ámbitos:

- **Petición:** El ActionForm solo estará disponible hasta el final del ciclo petición/respuesta. Una vez que se ha devuelto al cliente, la ActionForm y los datos dentro de esta dejan de ser accesibles.
- **Sesión:** De esta forma se mantienen los datos del formulario durante más tiempo que solo una petición. Es muy útil cuando por ejemplo creamos un asistente que captura los datos a través de múltiples páginas.

### 3.3.- LIBRERÍAS DE ETIQUETAS

---

Struts utiliza la librería de etiquetas JSP incluyendo varias categorías diferentes de etiquetas que ayudan a hacer la capa de presentación mucho más manejable y reutilizable.



Mediante las librerías de etiquetas los desarrolladores pueden interactuar con el resto del marco sin incluir código Java en las páginas JSP.

Las etiquetas JSP personalizadas mantienen su funcionalidad en una clase Java especial denominada el manejador de etiquetas, esta clase tiene acceso a todos los recursos JSP.

Las ventajas de usar etiquetas personalizadas son:

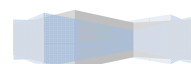
- Las etiquetas son reutilizables, ahorrando por lo tanto tiempo en desarrollo y comprobación.
- Las etiquetas se pueden personalizar utilizando atributos, tanto de forma estática como dinámica.
- Las etiquetas tienen acceso a todos los objetos disponibles para la página JSP, incluidas variables de petición, respuesta y resultado.
- Las etiquetas se pueden anidar, lo que permite interacciones más complejas dentro de una página JSP.
- Las etiquetas simplifican la legibilidad de una página JSP.

Las librerías de etiquetas incluidas con Struts son:

- **HTML:** Contiene etiquetas utilizadas para crear formularios de entrada de datos HTML, al igual que otras etiquetas utilizadas generalmente para la creación de interfaces de usuario basadas en HTML. La mayoría de estas etiquetas se debería anidar dentro de una etiqueta de formulario de Struts.
- **Bean:** Contiene etiquetas que se utilizan para acceder a JavaBeans y a sus propiedades asociadas, además de para crear nuevos beans.
- **Logic:** Contiene etiquetas que son de utilidad para gestionar la generación condicional de texto de salida, pasar un bucle sobre colecciones de objetos para una generación repetitiva de texto de salida y para la gestión de flujo de la aplicación.
- **Tiles:** Son etiquetas que sirven para crear plantillas y evitar repeticiones de código en las páginas JSP, dedicaremos un apartado a este tema mas tarde.
- **Nested:** Estas etiquetas sirven para anidar unas etiquetas dentro de otras.

## 4.- EL MARCO DE TRABAJO VALIDATOR

El marco de trabajo Validator, incluido en la distribución de Struts, permite configurar declarativamente rutinas de validación para una aplicación sin programar lógica especial de validación.



Aunque ya hemos hablado de la posibilidad de añadir lógica de validación dentro de la clase `ActionForm`, existen varios problemas con este enfoque:

- Codificar la lógica de validación dentro de cada `ActionForm` provoca redundancia en toda la aplicación.
- Si se necesita modificar o mejorar la validación que ocurre en un `ActionForm`, el código fuente se tiene que volver a compilar, por ello esto provoca un problema de rendimiento.

`Validator` permite mover toda la lógica de validación fuera de la `ActionForm` y configurarla de forma declarativa para una aplicación por medio de archivos XML. Por otro lado, es muy extensible ya que aunque proporciona algunas rutinas de validación, este marco se puede ampliar fácilmente.

Para hacer uso de `Validator`, se necesita utilizar el archivo de configuración *Validator-Rules.xml*. Este archivo contiene un conjunto global de reglas que cualquier aplicación puede configurar manualmente y que podrán ser modificadas si se desea ampliar o modificar las reglas que hay en él.

Además del archivo de configuración *Validator-Rules.xml*, es necesario crear el archivo *Validation.xml* en el que deben describirse las reglas de validación necesarias para cada `ActionForm`.

## 5.- EL MARCO DE TRABAJO TILES

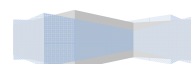
El marco de trabajo Tiles proporciona un mecanismo de plantillas mediante el cual separar las responsabilidades de diseño de las de contenido en las páginas de la aplicación, además de proporcionar las siguientes características:

- Posibilidades de uso de plantillas.
- Construcción y carga de página dinámica.
- Definiciones de pantalla.
- Soporte para reutilización de página y diseño.
- Soporte de internacionalización.
- Soporte de múltiples canales.

Antes de continuar vamos a explicar que es una plantilla, para entender mejor Tiles.

Una plantilla es una página JSP que utiliza una librería de etiquetas personalizada JSP para describir el diseño de una página. La plantilla actúa como una definición del aspecto que tendrán las páginas de una aplicación, sin especificar el contenido. El contenido se inserta en la página de plantilla en el momento de ejecución. De esta forma, una o más páginas pueden utilizar la misma plantilla.

Un tile es un área de trabajo o región dentro de una página Web. Una página puede contar con una o varias regiones.



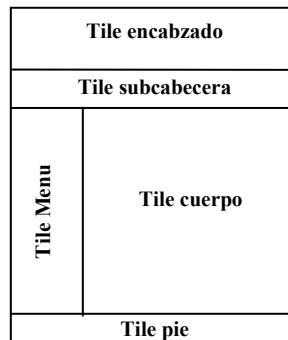


Figura 118: Ejemplo de Tiles en una aplicación Web

La mayoría de las veces, los atributos que usaremos en cada tile serán constantes en toda la aplicación, por ello será redundante especificarlos en cada tile. Lo más común es declarar los atributos en un solo lugar, y que los tiles los incluyeran allí donde los necesitasen.

Las definiciones de tiles proporcionan esta funcionalidad. Una definición permite especificar estáticamente los atributos que utiliza una plantilla, lo que a su vez permite, especificar solamente los atributos específicos de página en los tiles. Las definiciones permiten:

- Centralizar la declaración de descripción de página.
- Evitar la declaración repetitiva de páginas casi idénticas.
- Especificar el nombre de una definición como un forward en el archivo de configuración de Struts.
- ...

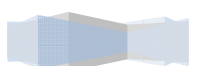
Las definiciones se pueden declarar en una JSP o en un archivo XML.

## 6.- CONFIGURAR APLICACIONES CON STRUTS

Struts utiliza dos tipos de archivos de configuración distintos, que se deben configurar adecuadamente antes de que una aplicación funcione correctamente. Ambos archivos están basados en XML.

El primero de ellos es el descriptor de despliegue de la aplicación Web denominado *web.xml*. Este archivo es necesario para cualquier aplicación Web aunque no haya sido creada con Struts, pero en él existe información de despliegue específica de Struts que se debe configurar cuando se crea una aplicación de este tipo.

El segundo archivo es el de configuración de Struts, conocido generalmente como *struts-config.xml*. En él se puede configurar de forma declarativa muchos de sus parámetros de aplicación.



## 6.1.- CONFIGURAR EL ARCHIVO WEB.XML

En primer lugar, debemos configurar la `ActionServlet` que recibirá todas las peticiones entrantes para la aplicación. Hay que realizar dos pasos para configurarla, el primer paso es utilizar el elemento `servlet` para configurar la instancia de servlet que se puede mapear más tarde en el elemento `servlet-mapping`.

Algunos elementos hijos de servlet son:

ELEMENTO	DESCRIPCIÓN
<b>servlet-name</b>	Especifica el nombre usado por el descriptor de despliegue para hacer referencia al servlet durante todo el archivo.
<b>servlet-class</b>	Indica la clase a la que pertenece el servlet.
<b>init-param</b>	Especifica los parámetros de configuración del servlet.

Tabla 19: Elementos hijos del elemento servlet

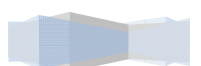
El siguiente paso es configurar el mapeado del servlet utilizando el elemento `servlet-mapping`.

Se muestra un ejemplo de configuración del servlet:

```
<servlet>
  <servlet-name>nombreDelServlet</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
  <init-param>
    <param-name>host</param-name>
    <param-value>localhost</param-value>
  </init-param>
  <init-param>
    <param-name>port</param-name>
    <param-value>7001</param-value>
  </init-param>
</servlet>

<servlet-mapping>
  <servlet-name>nombreDelServlet</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

Tras ello, debemos configurar las librerías de etiquetas JSP que vamos a usar en la aplicación. Para declarar una librería de etiquetas debemos utilizar el elemento `taglib`, que contiene dos elementos:



ELEMENTO	DESCRIPCIÓN
<b>taglib-uri</b>	Especifica una URI que identifica una librería de etiquetas.
<b>taglib-location</b>	Especifica la ubicación del archivo descriptor de librería de etiquetas.

**Tabla 20: Elementos hijo del elemento taglib**

Se muestra un ejemplo de configuración de librería de etiquetas:

```
<taglib>
<taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
<taglib-location>/WEB-INF/struts-html.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
<taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>
```

Por otro lado, si lo deseamos, debemos configurar el elemento `welcome-file-list`, que permite configurar recursos por defecto que se deberían usar cuando se facilita un URI válido pero parcial.

Se muestra un ejemplo de configuración de `welcome-file-list`:

```
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

Aunque el marco de desarrollo de Struts proporciona un mecanismo adecuado de gestión de error, en algunos momentos se pueden producir excepciones servlet o JSP que se envían al usuario. Para impedir que esto ocurra, debemos utilizar el elemento `error-page`.

Si se desea controlar una excepción JSP debemos indicar el código del error y la página a la que queremos dirigirnos cuando ocurra, por ejemplo:

```
<error-page>
  <error-code>404</error-code>
  <location>/common/404.jsp</location>
</error-page>
```

En cambio si se desea controlar una excepción de servlet, debemos indicar el tipo de excepción (`RuntimeException`, `ServletException` ó `IOException`) y la página a la que queremos dirigirnos cuando ocurra, por ejemplo:

```
<error-page>
  <error-type>javax.servlet.ServletException</error-type>
  <location>/common/system-error.jsp</location>
</error-page>
```



## 6.2.- CONFIGURAR EL ARCHIVO STRUTS-CONFIG.XML

El elemento `data-sources` permite establecer un origen de datos rudimentario que se puede utilizar dentro de Struts. Un origen de datos actúa como una fábrica para conexiones de BD y proporciona un único punto de control.

Este elemento puede contener cero o más elementos `data-source`, y cada elemento `data-source` puede contener múltiples elementos `set-property` mediante los cuales se especifica el nombre y valor de una propiedad de configuración JavaBeans adicional cuyo método `setter` se llamará en el objeto que representa el elemento de alrededor.

ELEMENTO	DESCRIPCIÓN
<b>className</b>	La clase de implementación del bean de configuración que albergará la información de origen de datos.
<b>key</b>	El atributo de contexto de servlet debajo del que se almacenará este origen de datos.
<b>type</b>	El nombre de la clase Java que implementa el origen de datos.

**Tabla 21: Atributos para el elemento `data-source`**

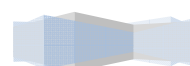
Se muestra un ejemplo de configuración para conectarse a una BD MySQL:

```
<data-sources>
  <data-source>
    <set-property property="autoCommit" value="true" />
    <set-property property="description" value="MySQL
    Data Source" />
    <set-property property="driverClass"
    value="com.jdbc.mysql.Driver" />
    <set-property property="url"
    value="jdbc:mysql://localhost:3360/Ejemplo" />
  </data-source>
</data-sources>
```

El elemento `form-beans` permite configurar múltiples clases `ActionForm`. Dentro de este elemento se pueden configurar cero o más elementos `form-bean`.

ELEMENTO	DESCRIPCIÓN
<b>className</b>	La clase de implementación del bean de configuración que albergará la información del form-bean.
<b>dynamic</b>	Si estamos utilizando Action dinámicos esta propiedad contendrá el valor <code>true</code> .
<b>name</b>	Identificador único para el bean durante todo marco.
<b>type</b>	El nombre de la clase Java que implementa el form-bean.

**Tabla 22: Atributos del elemento `form-bean`**



Se muestra un ejemplo de configuración de form-beans:

```
<form-beans>
  <form-bean
    name="tablasAuxiliaresForm"
    type="com.oreilly.struts.tablasauxiliares.Tablas
    AuxiliaresForm" />
</form-beans>
```

Dentro del elemento global-exceptions, se pueden configurar manejadores de excepción de forma declarativa.

```
<global-exceptions>
  <exception
    key="global.error.invalidLogin"
    path="/security/signin.jsp"
    scope="request"
    type="com.oreilly.struts.framework.exceptions.
    InvalidLoginException" />
</global-exceptions>
```

Toda acción que se ejecuta termina al reenviarse o redirigirse a una vista que suele ser una página JSP o HTML, así en vez de hacer referencia directamente a la vista, Struts utiliza el concepto de reenvío para asociar un nombre lógico al recurso.

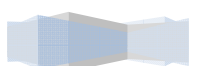
El elemento global-forwards permite configurar reenvíos que pueden ser usados por las acciones de la aplicación.

Se muestra un ejemplo de configuración de global-forwards:

```
<global-forwards>
  <forward name="operacionCorrecto"
    path="/comun/operacionCorrecto.jsp" redirect="true" />
  <forward name="operacionError"
    path="/comun/operacionError.jsp" redirect="true" />
</global-forwards>
```

El elemento action-mappings contiene un conjunto de cero o más elementos action para una aplicación Struts.

El elemento action describe un mapeado desde una ruta de petición específica a una clase Action correspondiente.





ELEMENTO	DESCRIPCIÓN
<b>className</b>	La clase de implementación del bean de configuración que albergará la información de acción.
<b>forward</b>	La ruta de acceso relativa a la aplicación para un servlet o recurso JSP al que se reenviará.
<b>name</b>	El nombre del form bean asociado a esta acción.
<b>path</b>	La ruta de acceso relativa a la aplicación para la petición enviada.
<b>scope</b>	Ámbito en el que se sitúa el form bean.
<b>type</b>	El nombre de la clase Java que implementa acción.
<b>validate</b>	Valor booleano que indica si el método validate() del form bean se debe llamar.

**Tabla 23: Algunos atributos del elemento action**

Se muestra un ejemplo de configuración de action-mappings:

```
<action
  path="/buscarTablasAuxiliares"
  type="com.oreilly.struts.tablasauxiliares.BuscarTablasAction"
  name="tablasAuxiliaresForm"
  scope="request"
  validate="false">
  <forward name="listaTablas"
    path="/jsp/tablasAuxiliares/buscarTablas.jsp"
    redirect="false"/>
</action>
```

El elemento message-resources, especifica las características del paquete de recursos de mensajes que contiene los mensajes localizados para una aplicación.

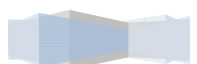
Se muestra un ejemplo de configuración de message-resources:

```
<message-resources
  parameter=" com.oreilly.struts.resources.ApplicationResources"/>
```

El elemento plug-in permite que las aplicaciones Struts descubran recursos dinámicamente cuando se inician.

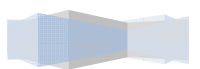
Para crear un plug-in, se necesita crear una clase Java que implemente la interfaz org.apache.struts.action.plugin y añadir un elemento plug-in al archivo de configuración.

Se muestra un ejemplo de la configuración del plug-in para el framework Validator:





```
<plug-in
  className="org.apache.struts.validator.ValidatorPlugIn">
  <set-property
    property="pathnames"
    value="/WEB-INF/validator-rules.xml,
          /WEB-INF/validator-rules-custom.xml,
          /WEB-INF/validation.xml"/>
</plug-in>
```



## ANEXO IV.- EL MARCO DE TRABAJO HIBERNATE

A continuación, se explicará más detalladamente como funciona Hibernate, junto con sus ficheros de configuración y los diferentes elementos que incluye.

### 1.- CONCEPTOS BÁSICOS DE HIBERNATE

Hibernate funciona asociando a cada tabla de la base de datos un Plain Old Java Object (POJO). Un POJO es similar a una Java Bean, con propiedades accesibles mediante métodos getter y setter.

Para poder asociar el POJO a su tabla correspondiente en la base de datos, Hibernate usa los ficheros de configuración de mapeo hbm.xml. En estos ficheros se declaran las propiedades del POJO y sus correspondientes nombres de columna en la base de datos, asociación de tipos de datos, referencias, relaciones n a n con otras tablas...

### 2.- EL FICHERO DE CONFIGURACIÓN DE HIBERNATE

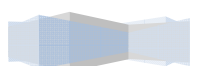
En el fichero *hibernate.cfg.xml* se determinan los aspectos relacionados con el gestor de base de datos y las conexiones a él.

Se muestra un ejemplo de fichero de configuración de Hibernate:

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-2.0.dtd">

<hibernate-configuration>
  <session-factory>
    <property name="connection.driver_class">
      com.mysql.jdbc.Driver</property>
    <property name="connection.url">
      jdbc:mysql://localhost/test</property>
    <property name="connection.username">yankoa</property>
    <property name="connection.password"></property>
    <property name="dialect">
      net.sf.hibernate.dialect.MySQLDialect</property>
    <property name="show_sql">true</property>
    <property name="transaction.factory_class">
      net.sf.hibernate.transaction.JDBCTransactionFactory
    </property>

    <!-- Mapping files -->
    <mapping resource="com/shop/Category.hbm.xml" />
    <mapping resource="com/shop/Product.hbm.xml" />
  </session-factory>
</hibernate-configuration>
```



Los cuatro primeros property contienen la configuración necesaria para la conexión JDBC que utilizara Hibernate, es decir, se indica la clase que contiene el driver para conectarse a la base de datos, url de la conexión y el nombre y contraseña para acceder a la base de datos.

Mediante la propiedad dialect se indica el nombre de la clase que se encargará de comunicarse con la base de datos en el lenguaje SQL que entienda la base de datos. Este parámetro ha de ser siempre especificado y el valor ha de ser una subclase que herede de `net.sf.hibernate.dialect.Dialect`.

En los siguientes property se indica que Hibernate delegará las transacciones a la conexión JDBC subyacente y después especificamos el proveedor de caché.

Finalmente se indica a través de mapping-resource el path donde se encuentran los ficheros de mapeo.

### 3.- EL FICHERO DE MAPEO DE HIBERNATE

Para poder asociar el POJO a su tabla correspondiente en la base de datos, Hibernate usa los ficheros de mapeo con el sufijo *hbm.xml*.

En estos ficheros se declaran las propiedades del POJO y sus correspondientes nombres de columna en la base de datos, asociaciones de tipos de datos, referencias, relaciones n a n con otras tablas...

Se muestra un ejemplo de un fichero de mapeo de Hibernate:

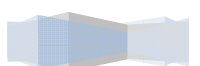
```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd">

<hibernate-mapping>
    <class name="escuela.beans.Categoria" table="categorias">
        <id name="id" type="integer" column="ID" unsaved-value="-1">
            <generator class="identity"/>
        </id>

        <property name="categoria" column="CATEGORIA"
            type="string"
            unique="true"
            not-null="true"/>
    </class>
</hibernate-mapping>
```

A continuación, describimos cada uno de los elementos que conforman el fichero de mapeo:

- **<hibernate\_mapping>**: Permite especificar, mediante atributos, diversas características como el esquema de la base de datos.



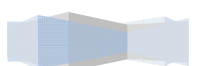
- **<class>**: Permite especificar el emparejamiento entre una clase java con una tabla de la base de datos. A través de name, especificamos el nombre de la clase Java, y mediante table, el nombre de la tabla de la base de datos con la que se empareja.
- **<id>**: Empareja una de las propiedades del objeto con la clave primaria de la tabla. A través de name, identificamos la propiedad del objeto que se empareja; con column, indicamos con que columna de la tabla se empareja; opcionalmente con type indicamos el tipo de datos con el que estamos trabajando y mediante unsave-value, opcionalmente indicamos el valor que debe tener el campo clave cuando todavía no ha sido almacenado el objeto en la base de datos.
- **<property>**: Empareja una de las propiedades del objeto con algún atributo de la tabla, que no sea clave primaria. Es muy similar a <id> pero tiene dos nuevas propiedades: unique que permite especificar si se admiten o no valores duplicados (es opcional) y not-null que indica si puede ser una propiedad nula o no.
- **<generator>**: Este subelemento permite definir cómo se generan los valores de las claves primarias, existen diversos métodos para ello, pero solo se comentarán los más utilizados:
  - **identity**: Utiliza el soporte de campos identity que ofrecen la mayoría de las bases de datos para generar la clave automáticamente.
  - **assigned**: Permite que sea la propia aplicación quien elija un identificador para el objeto antes de almacenarlo en la base de datos.
  - **sequence**: Utiliza una secuencia para generar la clave.
  - **hilo**: Utiliza una tabla auxiliar para generar las claves.
  - **native**: Permite que Hibernate escoja entre los métodos identity, sequence o hilo en función de las características del gestor de base de datos con el que se trabaje.

### 3.1.- MAPEAR ASOCIACIONES

---

Se ha mostrado como mapear un único objeto a la base de datos, es decir, relaciones uno a uno 1-1, pero en todo diseño relacional los objetos se referencian unos a otros a través de relaciones mas complejas como son : uno a muchos (1-n), muchos a muchos (n-m), muchos a uno (n-1).

Para poder explicar las asociaciones se utilizaran diversos ejemplos, con el fin de dejar más claro su uso.



### Asociaciones n-1

La relación “muchos a uno” necesita en la tabla un identificador de referencia, el ejemplo clásico es la relación entre padre - hijos. Un hijo necesita un identificador en su tabla para indicar cual es su padre. Pero en objetos en realidad no es un identificador si no el propio objeto padre, por lo tanto el componente n-1 es en realidad el propio objeto padre y no simplemente su identificador (aunque en la tabla se guarde el identificador).

Para reflejar este tipo de relación se utiliza el elemento `<many-to-one>` que contiene los siguientes atributos:

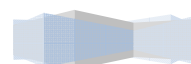
- **name:** El nombre de la propiedad.
- **column:** Columna de la tabla donde se guardara el identificador del objeto asociado.
- **class:** Nombre de la clase asociada.
- **cascade ("all|none|save-update|delete"):** Especifica que operaciones se realizaran en cascada desde el objeto padre. Este atributo es opcional.

Se muestra un ejemplo de una relación n-1:

```
<many-to-one name="padre" column="ID_PADRE" class="padreR"/>
```

### Asociaciones n-m

En las asociaciones “muchos a muchos” se contará con dos clases A y B, donde un elemento de la clase A tendrá un conjunto de hijos de la clase B, y un elemento de B tendrá otro conjunto igual o distinto de elemento A.



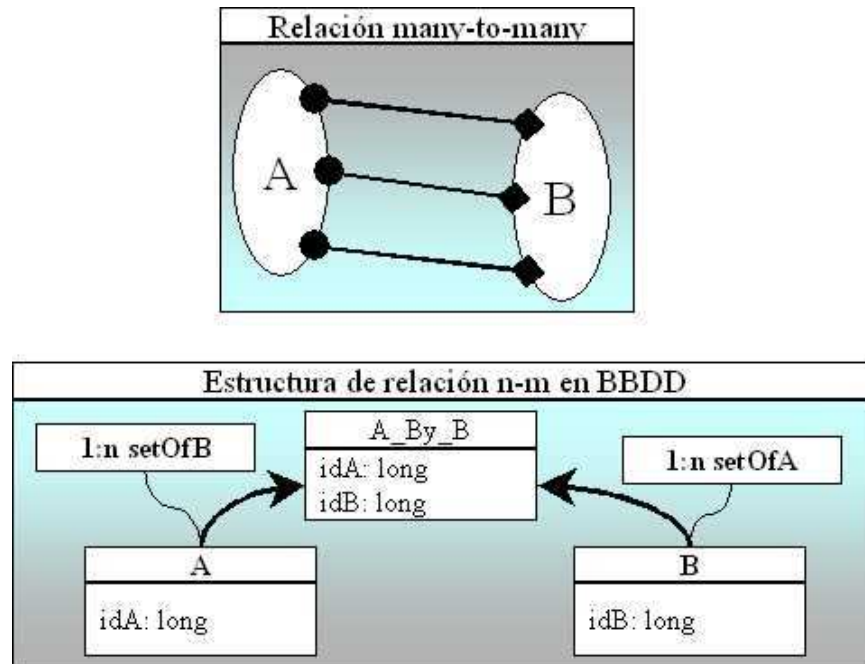
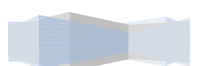


Figura 119: Relación n-m

Para reflejar este tipo de relación se utiliza el elemento <many-to-many> que contiene los siguientes atributos:

- **name:** El nombre de la propiedad.
- **table:** Nombre de la tabla de la colección.
- **column:** Columna de la tabla donde se guardara el identificador del objeto asociado.
- **class:** Nombre de la clase asociada.
- **key:** Columna clave que se relaciona con la clave primaria del objeto intermedio.
- **lazy ("true"|"false"):** Permite el uso de inicialización "lazy". Este tipo de inicialización hace que los objetos de la colección sean solicitados en demanda y no se carguen todos a la vez.
- **inverse:** Señala la colección como el fin de una asociación bidireccional.
- **cascade:** Permite las operaciones en cascada hacia los entidades hijas. Puede contener los siguientes valores:
  - **save-update:** Se aplica solo a las operaciones de inserción o modificación.
  - **delete:** Se aplica solo a las operaciones de borrado.



- **all:** Se aplica a cualquier operación.
- **none:** No se realiza ninguna operación en cascada.

**sort:** Especifica una colección con una ordenación natural o con una clase comparadora dada.

- **order-by:** Columna\’s de la tabla que definen el orden de iteración. Puede ser ascendente o descendente.

Para diseñar este tipo de asociaciones, se necesita crear una tabla intermedia que relacione los códigos de los elementos de A con los elementos de B. Por lo tanto se ha de mapear en una tabla a parte con las claves de las dos tablas como claves ajenas.

Esto se podría mapear de la siguiente manera:

```
<set role="setOfB" table="A_By_B">
  <key column="A_id"/>
  <many-to-many column="B_id" class="elementOfB"/>
</set>
```

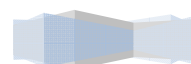
Como se puede observar, no existe una columna extra en B que diga los elementos de B que le corresponden a un elemento de A. En vez de eso se ha creado una tabla nueva A\_By\_B que contiene los pares de claves relacionados tanto de A hacia B como de B hacia A. Para que sea bidireccional tiene que ser declarada en el mapeo de la clase B como sigue:

```
<set role="setOfA" table="A_By_B" inverse="true">
  <key column="B_id"/>
  <many-to-many column="A_id" class="elementOfA"/>
</set>
```

### Asociaciones 1-n

Para crear relaciones “uno a muchos”, es necesario mapear correctamente las dos tablas que participan en la relación, creando por un lado la relación <one-to-many> y por otro una relación <many-to-one>.

Este tipo de relación puede contener los mismos atributos que las relaciones <many-to-many>.





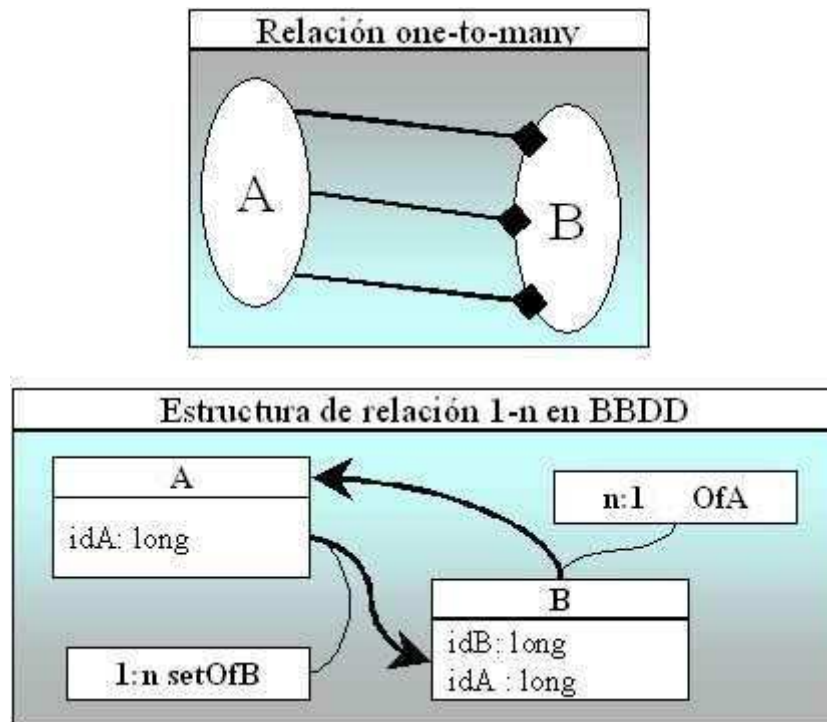


Figura 120: Relación 1-n

Una asociación 1-n de A hacia B requerirá un nuevo campo en B con el valor del índice de A al que se encuentra asociado. En la tabla A no será necesario ningún nuevo campo.

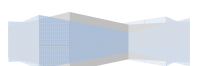
Se muestra un ejemplo de relación 1-n:

```
<set name="setOfB" table="B">
  <key column="A_id"/>
  <one-to-many class="B"/>
</set>
```

### 3.2.- LAS SESIONES

Para que se puedan utilizar los mecanismos de persistencia de Hibernate, el desarrollador debe mantener una conversación con el motor de Hibernate mediante un objeto especial, el objeto de “sesión”.

Se puede equiparar a grandes rasgos al concepto de conexión de JDBC y cumple un papel muy parecido, es decir, sirve para delimitar una o varias operaciones relacionadas dentro de un proceso de negocio, demarcar una transacción y además aporta algunos servicios adicionales como una caché de objetos para evitar interacciones innecesarias contra la BD. En este sentido, la clase Session ofrece métodos como *save(Object object)*, *createQuery(String queryString)*, *beginTransaction()*, *close()*, para interactuar con la BD tal y como se hace con una conexión JDBC con la diferencia de que existe una mayor simplicidad, es decir, guardar un objeto, por ejemplo, consiste en algo así como *session.save(miObjeto)*, sin necesidad de especificar una sentencia SQL.



Las sesiones son un concepto ligado a un proceso de negocio, por tanto es natural pensar que una sesión siempre va a pertenecer a un mismo hilo de ejecución, aunque técnicamente se pueden compartir sesiones entre hilos, esto no se debe hacer jamás por no ser una buena política de diseño y los consecuentes problemas que puede generar, es decir, en un entorno multiusuario y por tanto multihilo habrá múltiples sesiones simultáneas, cada una perteneciente a su correspondientes hilo y con su contexto de objetos en caché, transacciones, etc. Como tal no sorprende que las sesiones no son “thread-safe” y que la información vinculada a ella no sea visible para otras sesiones. Es también lógico que tenga que existir una “institución” superior para crear sesiones y realizar operaciones comunes a los diferentes hilos como lo puede ser la gestión de una caché compartida entre hilos o caché de segundo nivel. Este elemento es la clase `SessionFactory` y en ella podremos encontrar métodos como *`openSession()`* o *`evict(Class persistentClass)`*.

### 3.3.- HIBERNATE QUERY LENGUAJE: HQL

Hibernate ofrece un lenguaje de consultas que agrupa un potente y flexible mecanismo de consulta, almacenamiento, actualización y recuperación de objetos desde una base de datos.

Este lenguaje, que recibe el nombre de Hibernate Query Lenguaje (HQL), es una extensión orientada a objetos de SQL

El uso de HQL nos permite usar un lenguaje intermedio que según la base de datos que usemos y el dialecto que especifiquemos será traducido al SQL dependiente de cada base de datos de forma automática y transparente.

A modo de ejemplo se describen los pasos que sigue Hibernate para resolver una consulta sencilla como `FROM Persona P WHERE P.Nombre = 'Yankoa'`:

- 1) Determina el dialecto SQL que se esta utilizando a través del archivo de configuración de Hibernate.
- 2) Traduce la consulta al dialecto SQL que necesite.

```
SELECT Id, Nombre, Apellidos  
FROM Personas  
WHERE Persona.Nombre = 'Yankoa'
```

- 3) Ejecuta la consulta vía JDBC y obtiene un `ResultSet`.
- 4) Construye un objeto a partir de los elementos proporcionados en el `ResultSet`.

